This dissertation is submitted for the degree of *Doctor of Philosophy*

# Towards more reliable feature evaluations for classification

Gabriel Prat Masramon

Barcelona, 2015


Department of Computer Science

Supervisor: Lluís A. Belanche Muñoz

# Contents

# List of Figures

# List of Tables

# List of Algorithms

**Abstract**

In this thesis we study feature subset selection and feature weighting algorithms. Our aim is to make their output more stable and more useful when used to train a classifier. We begin by defining the concept of stability and selecting a measure to asses the output of the feature selection process. Then we study different sources of instability and propose modifications of classic algorithms that improve their stability. We propose a modification of *wrapper* algorithms that take otherwise unused information into account to overcome an intrinsic source of instability for this algorithms: the feature assessment being a random variable that depends on the particular training subsample. Our version accumulates the evaluation results of each feature at each iteration to average out the effect of the randomness. Another novel proposal is to make *wrappers* evaluate the remainder set of features at each step to overcome another source of instability: randomness of the algorithms themselves. In this case, by evaluating the non-selected set of features, the initial choice of variables is more educated. These modifications do not bring a great amount of computational overhead and deliver better results, both in terms of stability and predictive power. We finally tackle another source of instability: the differential contribution of the instances to feature assessment. We present a framework to combine almost any instance weighting algorithm with any feature weighting one. Our combination of algorithms deliver more stable results for the various feature weighting algorithms we have tested. Finally, we present a deeper integration of instance weighting with feature weighting by modifying the Simba algorithm, that delivers even better results in terms of stability.

Chapter

# 1 Introduction

This thesis deals with the so-called **feature subset selection** (FSS) and **feature weighting** (FW) problems in supervised inductive learning scenarios. The generic purpose of FSS is the improvement of the inductive learner, either in terms of learning speed, generalization capacity or simplicity of the representation by identifying the features that are of interest for the learning purposes. A FSS algorithm should be able to identify the optimal subset of features containing all the *strongly relevant* features (always necessary for an optimal subset) and a minimal subset of the *weakly relevant* (needed for class discrimination in some situations) without any *redundant* (features whose information is subsumed by some other feature or group thereof) or *irrelevant* (not useful for the learner) features. FW, on the other hand, aims at assigning weights to features proportional to their importance.

Traditionally, the main goal of FSS techniques in the classification case is to improve predictive accuracy. Bearing in mind that the values of the criterion guiding the quest for the best feature subset are realizations of a random variable, we claim that there is another major challenge in both FSS and FW: model **stability**. This is particularly important in greedy wrapper sequential FSS methods because the decision as to which feature should be preferred at each step involves uncertainty. A different feature choice at an early step may completely change the search path and lead to a substantially different set of features, as it often happens. In some real-world application domains (such as biomedical ones), *stability* of the selected or weighted features may be of paramount importance.

The focus of the thesis is to **measure**, **study** and eventually **improve the stability** and **accuracy** of a classifier trained with the results obtained by FSS and FW processes. We do so from a variety of perspectives and provide several modifications to well-known *filters* (using a model-independent criterion based on prior knowledge of the data) and *wrappers* (using a specific classifier to assess the usefulness of the feature sets) methods. All of our improvements aim to **maximize the usage of the available data** at each step of the process.

### 1.0.1 Characterization of the stability problem

Ill-posed problems are problems for which a solution is either non-uniquely determined, non-existent or *unstable* under data perturbations. Typical examples of ill-posed problems abound in mathematics, such as integral equations of the first kind and many inverse problems, like systems of linear equations [77]; these problems are frequently encountered in science and engineering. The term itself was introduced by Hadamard who investigated problems in mathematical physics.

In this thesis the focus is in the stability of FSS algorithms, where we have identified several *sources of instability*:

1. The algorithms are run on a random sample; by changing the sample different results will be obtained.
2. The evaluation measure is computed in different views of the random sample (this is known as *resampling*); by changing the view different results will be obtained.
3. The algorithm may have a randomized aspect as part of its execution: e.g. the RELIEF algorithm chooses observations at random from the sample; different executions will yield different results.
4. The algorithm may be unstable in nature in the above sense: in the simplest situation, the addition or elimination of a single observation to/from the sample may yield different results;
5. There is an inherent lack of relevant features among those measured in the data sample.

Another common assumption (at least, an implicit one) is that classification accuracy will change smoothly with changes in the training data. This is typically not the case, particularly in small sample problems, where the presence or absence of certain data points can have a large impact on model generation, selection and assessment. In the end, all causes except the last one are amenable to study and ultimate control, which provides support for the goals of this thesis. Note that in our view the outcome of a FSS algorithm is a *random set*, a random vector composed of non-independent binary variables.

A key issue is found in the fact that different data points contribute differently to the importance of features. Those that contribute a lot will positively contribute to the obtention of more unstable solutions. Therefore a key issue is to be found in the interplay between the method used for computing the observation weights and that used for computing the feature weights. Traditionally, these two processes have been treated as separate and there are several approaches in the literature; however, they are not independent and hence they should be treated in a more unified way. Last but not least, the chosen learner could eventually make use of both sets of weights to fit the data giving more importance to some features *and* to some data points.

In summary, the ideas developed in this thesis ultimately advocate that getting additional sources of information from the available data brings benefits in the stability of the obtained models, many times without paying a toll in accuracy. This document is structured in chapters focusing on specific points of view regarding FSS and FW stability and classification accuracy of a learner trained with the resulting feature subset or feature weights.

### 1.0.2 Hypothesis to be tested

Here we break the main goal in this thesis (i.e. having more reliable feature evaluations in supervised learning) into different building blocks and formulate some hypotheses for each one that may contribute to the overall goal. We have structured these building blocks according to which of the *sources of instability* described above they are addressing. These building blocks include the different parts of the FSS process:

1. The way instances are selected, weighted and used
2. The way features are selected
3. The way selected feature sets are evaluated; this includes their ability to deal with irrelevant and redundant features and the use made of the information that is being computed along the process
4. The outcome of the process, which can be selecting a subset of features, weighing features or a combination of weighting instances and features

#### Random samples

The first *source of instability* described above was the one introduced by running the FSS algorithms on different subsamples (e.g., in cross-validation partitioning). One of the topics we tackle in this work is to study the existence of a relationship between feature subset redundancy and the stability of the resulting subsets. The motivation for this hypothesis is the following: according to [34], there is a subset of features (the *strongly relevant*) that should always be included in the selected subset; some of them (the *weakly relevant*) should only be included in certain situations. Our hypothesis here is that each of these *weakly relevant* features should only be included if is not *redundant* to the rest of the selected features. The problem is that since feature redundancy indicates that there is a certain degree of overlap among the information some features provide about the class we are trying to predict, there is no unambiguous way of choosing them. As an example, consider the extreme case of two identical features. If the aim of FSS is to find a minimal subset of features with maximum predictive power about the class, choosing both of them would not be optimal. Yet the decision on which to choose is absolutely arbitrary. In real life problems we will rarely find two identical features but one can find subsets of features that are almost equivalent.

With this scenario we will test the following **hypothesis: a larger amount of feature redundancy leads to more unstable results**.

Chapter 4 introduces theoretical work on redundancy: a Markov Blanket based measure that should indicate the index of redundancy a feature has with a given feature set. This can afterwards be useful to study the effects redundancy has on FSS and FW stability. We also provide initial theoretical work on feature importance and how it related to FW stability.

#### Intrinsic instability

Here we analyze a different approach to intrinsic instability: the effect of *irrelevant* features. In some cases by adding more and more of these features we can introduce intrinsic instability.

Consider taking the popular FW algorithm Relief. This algorithm selects features that help discriminate *near* instances using the average of feature differences to compute the distances; let us consider again an extreme case. Imagine we have thousands of irrelevant features and only a few relevant ones, which could be a quite realistic scenario in some biomedical settings. Our **hypothesis: using all the features for distance calculation may lead to arbitrary distances between features when the number of *irrelevant* features is vey high**. The *relevant* features may make a very small contribution compared to the vast number of *irrelevant* ones. Hence, adding more and more *irrelevant* features can mislead Relief and, in the end, render it almost useless.

We analyze this in Chapter 5 and propose some modifications to overcome this limitation. We analyze different distance metrics in addition to the original one and study the effect they have in accuracy, redundancy detection and stability. We also test ways to use the weights computed at each step to influence the distance calculation as the Simba algorithm does. We propose a novel modification to both algorithms to improve their stability consisting in incrementing the feature weights' contribution to distance calculations over time to minimize the impact of randomly choosing the first instances.

An additional example is found in sequential wrapper FSS algorithms. These algorithms use an inducer to assess the feature subset quality at each iteration and add (or subtract) one feature at each iteration, completely forgetting about the feature assessments done in previous iterations. Consider, for example, an algorithm that sequentially subtracts features. In the early iterations we are evaluating most of the features. Assume that feature $x_i$ has survived until the $n$-th iteration. At this point $x_i$ has been evaluated $n$ times together with the rest of the features so we have some good insights on whether $x_i$ was useful when combined with the previous subsets of features. Our **hypothesis: taking into account the *accumulated* evidence about the features in previous iterations will deliver better and more stable results**. Knowing that feature assessment is a random variable, taking various values (the evaluations in all the previous iterations) into account the results will tend to average leading to more stable results.

In Chapter 7 we show that by modifying existing sequential feature selection algorithms to take the accumulated evidence into account, the resulting algorithms do improve in classification accuracy and stability.

### Random parts of the algorithm

A fundamental characteristic of sequential FSS (SFSS) algorithms is that at each step of the process they evaluate a subset of features. In the case of sequential forward selection algorithms, they start with an empty set of features and add one feature to this set at each iteration. Thus, at early stages, these algorithms do not take into account the intricate relationships that may exist among features. In particular, at the first iteration, they are evaluating each feature on their own. In problems with a high grade of dependency among features this should not be a good idea. Let us devise again an example to demonstrate how this leads to instability (and to lower performance). Assume we have a three-feature problem like the one in Table 1.1 below:

In this problem we start evaluating each of the 3 features and every one of them is correlated 3/4 of the time with the class. With this setup every feature may be given the same

| $f_1$ | $f_2$ | $f_i$ | $C$ |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 |

**Table 1.1:** Two relevant features and one irrelevant: $C = f_1 \wedge f_2$

importance when evaluated separately. Thus a forward SFSS could select any of them at random being this random selection the third *source of instability* described above. We can also see that, being $f_i$ a constant feature, it will never add any value when combined with one of the other two; while combining $f_1$ and $f_2$ leads to a 0% prediction error. Yet the perfect subset $\{f_1, f_2\}$ would never be found by a SFFS algorithm that chose $f_i$ in the first iteration. The effects of bad choices in early iterations can dramatically affect stability of the chosen subsets. Generalizing this problem leads to our next **hypothesis: evaluating both the selected and the remainder set of features will lead to better and more stable results**. By evaluating both sets of features –(quasi)individual contributions and (quasi)full set interactions– we are taking two different perspectives of the same variable into account, and this extra information may lead to more informed and stable results.

In Chapters 6 we explore this idea and give support to the hypothesis.

#### Different data points contribute differently to stability

The last *source of instability* we explore in this thesis is the study of the effect that different data points have to the final result of FSS. As mentioned above, another of the *sources of instability* might come for the algorithm itself to be unstable in nature. As an example of what is meant, the addition of a single observation could lead to completely different results. Our **hypothesis: ignoring *outlier* data-points will lead to more stable solutions**. In fact, we will explore a soft version that consists in giving different importances to each data point according to some criteria. We will use the margin based criteria presented in [29] and a novel instance weighing measure that takes into account the distance of the instance to the hypothesis margin in a similar way that Relief does in order to improve existing FW algorithms. In Chapter 8 we outline a novel way to improve stability of both Relief and Simba. We propose to use the information about the different contribution each instance carries to feature importance detection. In fact, again in real-world domains, there may be outlying instances that should downweighted; for example, in trying to predict cancer using DNA microarray expression data, we should give less credit to instances corresponding to people exposed to high levels of radiation even if we have no information about the exposure. In this sense, **instance weighting** (IW) methods try to identify these instances and assign them lower weights. Various authors have worked on IW schemes in order to improve FSS but there is not much previous work on the **combination of IW with FW** in a synergistic way [54, 53]. We outline different methods to combine IW with FW, from the use of instance weights to drive the random instance choice at every step of FW to an embedded use of instance weights inside Simba's distance calculation. We have tested our new algorithms against the original counterparts using a variety of data sets, ranging from classical UCI data

to microarray datasets; we report results showing improved stability without loosing –and sometimes even improving– the classification accuracy.

## 1.1 Document structure

We start by reviewing the current state of the art in Chapter 2. We review different well known FS and FW algorithms and highlight their properties. Afterwards we provide an exhaustive review of FSS stability measures. Finally, we introduce the concept of hypothesis margin and describe two different studies that used this concept. One for FW – the Simba algorithm – and the other one for IW – margin based instance weighting (MBIW).

The following set of chapters start with some common experimental settings across all the experiments being conducted –Chapter 3. Then Chapters 4 to 8 explore the different parts of the stability problem as described above. To avoid giving too much information, only summaries of the results are displayed and discussed in the chapters. All the detailed experimental results are to be found in Appendices A to C.

Finally, the conclusions and avenues for future work are presented in Chapter 9.

## 1.2 Main contributions

In this thesis an in-depth study of FSS and FW stability has been performed. We study stability from different points of view to be able to detect the sources of instability and propose novel ways to deal with it. Our main contributions aresummarized as:

**A review of FSS stability measures.** We have performed an exhaustive study of the state of the art and compare the different properties of the proposed measures both from a theoretical and from an experimental points of view. We also highlight the weaknesses of some of the measures –e.g. some of them not having correction by chance. This is a great weakness as the measure will always favor solutions with either very few (or almost all) features in the selected set. With this study of the stability measures we select the one that overcomes all the weaknesses that will be used to assess the presented experimental work.

**A study of the relationship between stability and feature redundancy.** A novel theoretical definition of redundancy based on Markov Blankets is presented. With this definition we are able to describe one of the possible sources of instability and note that some of the classic feature selection algorithms such as Relief do not deal well with feature redundancy.

**A definition of feature importance.** A novel theoretical definition of feature importance and initial results on its stability are presented, based on its bias/variance decomposition.

**Improvements of *wrapper* FSS algorithms** We propose various novel ways of improving classic algorithms, both for feature selection and feature weighting, guided by the above hypotheses. We present a modification of FSS *wrappers* aimed at reducing the random

parts of the algorithm by evaluating both the selected and the remainder set of features, and prove it to be more stable –even slightly better in performance– when used to train an classifier. Another modification for *wrappers* is presented that makes them better at selecting features by accumulating all the evaluations of a given feature over time. This modification will produce feature subsets leading to better classification performance when used to train a classifier.

**A unified framework for instance and feature weighting.** To test our hypothesis that different data points contribute differently to FSS and FW stability we have provided a framework to combine FSS and FW algorithms with IW algorithms that assess the importance each instance should have in the feature selection process.

**Stability improvements for *filters*.** Using the above framework we improve the stability of five classic FW algorithms, with a special focus on one of them: SIMBA. We improve this *filter* algorithm by weighing the contribution of the instances in distances calculations with the weights obtained in the IW phase. This modification improves both stability and predictive power of the resulting feature set.

It is worth noting that all of our improvement proposals can be put in practice with almost no extra cost, unlike some of the most popular stability improvement proposals –like ensemble FSS. Moreover, the improvements in classification accuracy are achieved most of the times without selecting larger feature sets.

Chapter

# 2 State of the art

This chapter presents a review of the current state of the art in feature subset selection and feature weighting. A special emphasis is made to review current work on stability of feature subset selection algorithms. We study both the different approaches to define and measure stability and also the different proposals different authors have made to improve it. Finally we outline the existing definitions of *instance margins* (e.g. the distance of an instance to the decision boundary of a classifier) as we will use this concept to build more stable algorithms throughout the following chapters.

## 2.1 Feature subset selection

Traditionally, feature selection research has focused on searching for relevant features. Based on a review of previous definitions of feature relevance, John, Kohavi, and Pfleger classified features into three disjoint categories, namely, *strongly relevant*, *weakly relevant*, and *irrelevant features* [34]. Let $Y$ be a certain feature set, $C$ a class attribute, $x_i$ a feature in $Y$ and $X_i = Y \setminus \{x_i\}$.

**Definition 2.1** (Strong relevance)**.** $x_i$ is strongly relevant when

$$P(C|x_i, X_i) \neq P(C|X_i) \tag{2.1}$$

**Definition 2.2** (Weak relevance)**.** $x_i$ is weakly relevant when

$$\begin{aligned} P(C|x_i, X_i) = P(C|X_i), \text{ and} \\ \exists X'_i \subset X_i, \text{ such that } P(C|x_i, X'_i) \neq P(C|X'_i) \end{aligned} \tag{2.2}$$

**Definition 2.3** (Irrelevance)**.** $x_i$ is irrelevant when

$$\forall X'_i \subseteq X_i, P(C|x_i, X'_i) = P(C|X'_i) \tag{2.3}$$

With the above definitions, a feature is relevant if it is either strongly or weakly relevant; otherwise, it is irrelevant. The authors defined the solution to the feature selection problem to be the subset containing all of the *strongly relevant* features, a subset of the *weakly relevant* and none of the *irrelevant*. However, there is no clue on which of the *weakly relevant* features one should include in the final subset and which of them to discard.

Another definition of FSS that overcomes the above ambiguity problem is to see feature subset selection in a set $Y$ of size $n$ as a *search problem* where the search space is the power set of $Y$, $\mathcal{P}(Y)$ [46]. Without loss of generality, we assume that the evaluation measure $\mathcal{L} : \mathcal{P}(Y) \to \mathbb{R}^+ \cup \{0\}$ is to be maximized. The criterion $\mathcal{L}$ may be problem-independent or may be the classifier that will be used to solve a classification problem. In any case, we will refer to $\mathcal{L}(X)$ as the *usefulness* of feature subset $X$.

**Definition 2.4** (Feature Selection)**.** Let $\mathcal{L}$ be an evaluation measure to be optimized (say to maximize). The selection of a feature subset can be made under two premises:

- Find $X^* \subset Y$, such that:
$$X^* = \underset{X \in \mathcal{P}(Y)}{\arg\max}\, \mathcal{L}(X) \tag{2.4}$$
- Set a real value $\mathcal{L}_{min}$, this is, the minimum $\mathcal{L}$ that is going to be accepted. Being $X_k$ a subset of $Y$ with exactly $k$ attributes; find the $X_k \subseteq Y$ with smaller $k$ such that $\mathcal{L}(X_k) \geq \mathcal{L}_{min}$. Alternatively, given $\epsilon > 0$, find the $X_k \subseteq Y$ with smaller $k$, such that $|\mathcal{L}(X_k) - \mathcal{L}(Y)| < \epsilon \mathcal{L}(Y)$.

Notice that, with this definition, the optimal subset of features always exists but is not necessarily unique. Also noteworthy is the fact that, denoting by $X^*$ one of the optimal solutions, either of $\mathcal{L}(X^*) > \mathcal{L}(Y)$, $\mathcal{L}(X^*) = \mathcal{L}(Y)$, $\mathcal{L}(X^*) < \mathcal{L}(Y)$ may occur.

Ideally, feature selection methods search through all the subsets of features and try to find the best one. But it is clear to see that if we had to test all possible subsets, using either of the methods, of features we would have a combinatorial explosion. If our initial set of features is $Y$ and $|Y| = n$, the number of evaluations we would have to do would be equal to the cardinality of the power set of $Y$: $|\mathcal{P}(Y)| = 2^n$. A *complete* search (as with the *branch and bound* method), is a possible procedure to guarantee the finding of an optimal subset; this method also requires the monotonicity of the inducer evaluation. This implies that when a feature is added to the current subset, the value of the criterion or evaluation function does not decrease. In most practical applications this approach is computationally prohibitive and the mainstream of research on FSS has thus been directed to *sequential* suboptimal search methods.

A *sequential feature selection algorithm* (SFSA) is a polynomial-time computational solution that is motivated by a certain definition of *usefulness*. An important family of SFSAs perform an explicit search in the space of subsets by iteratively adding and/or removing features one at a time until some stop condition is met. These methods typically share the same basic steps as seen in Fig. 2.1:

1. The *subset generation* to produce candidate subsets for evaluation

2. The *evaluation criterion* providing the usefulness of each subset
3. The *stopping criterion* to decide when to stop
4. The *result validation* by prior knowledge or statistical tests

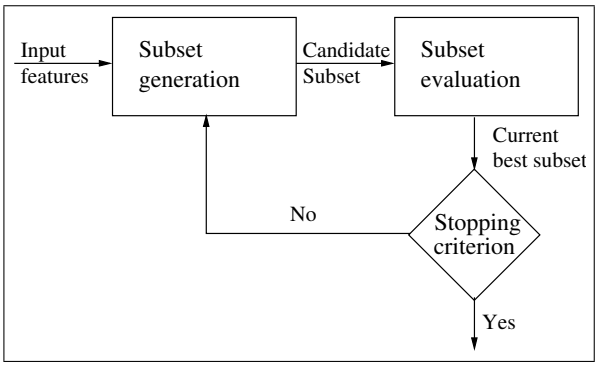



**Figure 2.1:** Feature selection framework

Looking at the evaluation criterion, John, Kohavi, and Pfleger [34] divided the feature selection methods into two main approaches: *filter* methods and *wrapper* methods. These two families of methods only differ in the way they evaluate the candidate sets of features. A third group of methods called *embedded* methods are a more recent approach to feature selection where the selection process is done implicitly as part of the classifier design.

**Filters**

Use a problem independent criterion, the basic idea of the filter methods is to select the features according to some prior knowledge of the data. For example, selection of features based on the *conditional probability* that an instance is a member of a certain class given the value of its features [4]. Another criterion commonly used by filter methods is the *correlation* of a feature with the class, i.e. selecting features with high correlation [28]. A well known filter algorithm is RELIEF [38, 43] that estimates the usefulness of features according to how well their values distinguish between the instances of the same and different classes that are near each other.

**Wrappers**

Suggest a set of features that is then supplied to a classifier, which uses it to classify the training data and returns the classification accuracy or some other measure thereof [41]. The search is guided by the classifier used as a *black box* (i.e. the feature selection process does not depend on how the classifier works). It is suggested in literature that wrapper methods, although they tend to overfit, perform better than filters [34, 39, 58] because using the classifier error rate used as the evaluation criterion catches better the structure and properties of the classifier. Among the proposed algorithms for attacking this problem are the sequential forward generation (SFG) and sequential backward generation (SBG), the *plus* $l$ *- take away* $r$ or $PTA(l, r)$ proposed by Stearns

[75] or the floating search methods [57]. They both introduce methods for the generation of the sets of features by combining steps of SFG with steps of SBG but keep using a certain $\mathcal{L}(X)$ as evaluation criterion.

**Embedded**
The idea is to optimize the evaluation criterion $\mathcal{L}(\cdot)$ directly and to perform feature selection as part of the classifier training. This mechanism can be found in algorithms like SVM [10], Adaboost [66], or CART [8].

Filter measures (e.g., probabilistic *separability* measures) do not induce the same preference order as would be obtained by comparing classification *error rates*. This is due to the fact that error rates not only capture class separability but any structural error imposed by the form of the classifier. As the second aspect is not reflected in FSS when based exclusively on filter measures, the resulting features may perform poorly when applied as the input to the classifier. Therefore, the legitimate way of evaluating feature subsets must be through the error rate of the classifier being designed [41].

**Algorithm 2.1:** SFG

1: $X_0 \leftarrow \emptyset$ {Initial subset}
2: $i \leftarrow 0$
3: **repeat**
4: $\quad \mathbf{S}_{i+1} \leftarrow \{X \mid X = X_i \cup \{x\} \wedge x \in Y \setminus X_i\}$ {Subset generation}
5: $\quad X_{i+1} \leftarrow \arg\max_{X \in \mathbf{S}_{i+1}} J(X)$ {Subset evaluation}
6: $\quad i \leftarrow i + 1$
7: **until** $J(X_i) \leq J(X_{i-1}) \vee i = n$ {Stopping criterion}
8: **if** $J(X_i) \leq J(X_{i-1})$ **then**
9: $\quad$ **return** $X_{i-1}$
10: **else**
11: $\quad$ **return** $X_i$ {Selected subset}
12: **end if**

**Algorithm 2.1** and **Algorithm 2.2** describe two of the classic feature selection algorithms using this point of view: sequential forward generation (SFG) and sequential backward generation (SBG). In these algorithms $X_0$ is the starting set of features of the algorithm, $\mathbf{S}_k$ the set of sets of features generated during the subset generation phase and $X_k$ the selected set of features at iteration $k$. It can be seen that the subset evaluation phase in the two algorithms is exactly the same while the initialization and the subset generation phases change.

## 2.2 Feature weighting

A different technique for the determination of feature usefulness is feature *weighting* (or feature ranking). It works by assigning a numeric value to each feature so as to indicate the feature's differential importance for predicting the class. Feature weighting can help solving the problem of feature selection. One possible approach to feature selection using feature weighting would be to assign weights to features and then choose features according

**Algorithm 2.2:** SBG

$X_0 \leftarrow Y$ {Initial subset}
$i \leftarrow 0$
**repeat**
  $\mathbf{S}_{i+1} \leftarrow \{X \mid X = X_i \setminus \{x\} \wedge x \in X_i\}$ {Subset generation}
  $X_{i+1} \leftarrow \arg\max_{X \in \mathbf{S}_{i+1}} J(X)$ {Subset evaluation}
  $i \leftarrow i + 1$
**until** $J(X_i) \leq J(X_{i-1}) \vee i = n$ {Stopping criterion}
**if** $J(X_i) \leq J(X_{i-1})$ **then**
  **return** $X_{i-1}$
**else**
  **return** $X_i$ {Selected subset}
**end if**

to their sorted weights. This can be done either by having a rule to binarize the weights, e.g. select all the features with weight greater than a certain threshold, or by a search favouring the evaluation of subsets containing features with greatest weight values. In fact, feature selection could be seen as a specific kind of feature weighting where the weights assigned to features are binary. We will explore various methods of existing feature weighting algorithms than and will discuss their properties. This section will review some of the most used feature weighting algorithms. Although the section is focused on feature weighting, most of the methods described below can also be used for feature selection.

On following subsections $S$ will be a dataset of $\{\mathbf{x_1}, \cdots, \mathbf{x_N}\}$ instances and $Y$ will represent the sets of features. $X$ or $X_i$ are possible subsets of features from $Y$. $\mathbf{x}$ or $\mathbf{x_1}$ are specific instances in $S$. $C$ represents the set of possible class values. And their lower case versions represent single value in its correspondent upper case set, e.g. we will use $c \in C$ and $x \in X$. We also will use a short notation to express probabilities, e.g. will write $P(x)$ to represent the probability for feature $X$ to have value $x$ or $P(c|x)$ to express the conditional probability of the class to have value $c$ knowing that the feature $X$ has value $x$.

### Conditional Probabilities based methods

The first group of methods we will look at are the ones based on conditional probabilities of class given a feature value. Two simple methods using this idea were introduced in [14]: per-category feature importance and cross-category feature importance (or, in short, PCF and CCF). One important limitation is that they can only deal with binary features, so numerical features must be discretized and symbolic features converted to a group of binary features. The weights assigned to features in the case of PCF depends on the class of the feature as seen in Eq. 2.5

$$w_{PCF}(X, c) = P(c|x), \text{ where } x \text{ would be the } \textit{positive} \text{ feature value} \tag{2.5}$$

so we have a weight for each feature and class. CCF relies on the same idea but instead of having one weight for each feature and class it have only a weight per feature. It does so by

averaging the weights across classes. In fact, as it shows Eq. 2.6, it uses the summation of squares of conditional probabilities.

$$w_{CCF}(X) = \sum_{c \in C} P(c|x)^2, \text{ where } x \text{ would be the } \textit{positive} \text{ feature value} \tag{2.6}$$

Later on [52] showed that PCF is too sensitive to class proportions and tends to answer the most frequent class when using it for classifying.

A more sophisticated approach that also makes use of conditional probabilities is the one used by the value difference method (VDM) introduced by [74]. This time no binarization of features is required, although numeric features still have to be discretized in order to calculate conditional probabilities as shown in Eq. 2.7. In addition this method does not assign weights to each feature but to each value of each feature.

$$w_{VDM}(X, x) = \sqrt{\sum_{c \in C} \left( \frac{P(x|c)}{P(x)} \right)^2} \tag{2.7}$$

This weighting scheme was originally used to calculate distances between features.

Finally we have Gini-index gain [8] in Eq. 2.8 which can be interpreted as the expected error rate

$$GG(X) = \sum_{x \in X} P(x) \sum_{c \in C} P(c|x)^2 - \sum_{c \in C} P(c)^2 \tag{2.8}$$

and is proven to be biased towards multiple valued features. In further sections we will see that this particular measure has some relation with the Relief algorithm.

### Information theory based methods

Not all the feature weighting methods are based on conditional probabilities, though. Now we will describe some methods based on information theory [68, 69].

The first one is just using Shannon's mutual information (MI) between two features $X$ and $Y$ in Eq. 2.9,

$$MI(X, Y) = H(X) - H(X|Y) = \sum_{x \in X, y \in Y} P(x, y) log_2 \frac{P(x, y)}{P(x)P(y)} \tag{2.9}$$

which is defined using entropies and conditional entropies (see Eq. 2.10),

$$\begin{aligned} \text{Entropy:} \qquad & H(X) = -\sum_{x \in X} P(x) \log_2 P(x) \\ \text{Conditional entropy:} \qquad & H(X|Y) = H(X, Y) - H(Y) \\ \text{Joint entropy:} \qquad & H(X, Y) = -\sum_{x \in X, y \in Y} P(x, y) \log_2 P(x, y) \end{aligned} \tag{2.10}$$

to weight features. A more informal but maybe more intuitive definition of mutual information is that MI measures the information of $X$ that is also in $Y$. If the features are independent

no information is shared so mutual information is zero. In the other end we have that one feature is an exact copy of the other, all the information it contains is also shared by the other so the mutual information is the same as the information conveyed by one of them, namely its entropy. A very popular feature weighting method uses the idea of mutual information. It was proposed by [33] and it is used in [60] when splitting nodes in top-down induction of decision trees (TDIDT) –best known as ID3. The term information gain (IG) in Eq. 2.11 is used there. Its intuitive interpretation would be: the more a feature reduces class entropy when knowing its value, the larger its weight. This is just another way to say that the more information is shared between an feature and the class, the larger its importance. Hence, if we have a set of classes $C$ we can define IG for the class knowing the value of a feature $X$ as shown in Eq. 2.11

$$IG(C|X) = MI(C, X). \tag{2.11}$$

Later on, similar methods were introduced to reduce the bias of IG towards features with a large number of values. The extreme case is found using an feature with an ID code. It is clear that knowing the ID code we can precisely know the class of any instance in our training set. The problem is that one could say nothing about a new instance –which will likely have an unseen ID code. One of these methods is gain ratio (GR) in Eq. 2.12, used by the C4.5 decision tree induction algorithm [59]. This method normalizes IG by the amount of information needed to predict a features value (the entropy of the feature). There are various other proposals, among them we find an entropic distance [48] in Eq. 2.13, and the de Mántaras distance between the class and the feature –Eq. 2.14– which was proved to be unbiased towards multiple-valued features.

$$GR(C|X) = \frac{IG(C|X)}{H(X)} \tag{2.12}$$

$$D_H(C, X) = H(C, X) - MI(C, X) \tag{2.13}$$

$$D_M(C, X) = \frac{H(X|C) + H(C|X)}{H(C, X)} = 2 - \frac{H(X) + H(C)}{H(C, X)} \tag{2.14}$$

### Distribution distance based methods

Another way to find dependencies between a feature and the class is to measure differences between their distributions. Perhaps the simplest way to do so is to compute the difference between the joint and the product distributions as shown in Eq. 2.15

$$\text{Diff}(C, X) = \sum_{c \in C, x \in X} |P(c, x) - P(x)P(c)| \tag{2.15}$$

and this distance can be directly used as the features weight. Large differences between the joint and the product distributions indicate large dependency of the class on the feature, so the feature should be given a large weight. This can easily be applied to continuous features changing the sum for an integral. It can also easily be rescaled to the $[0, 1]$ interval as it has an upper bound of $1 - \sum_{x \in X} P(x)^2$.

More distance functions can be used here. An interesting one is the Kullback-Leibler divergence –which is not actually a metric distance as it is not symmetric, i.e., $D_{KL}(X||Y) \neq$

$D_{KL}(Y||X)$. The application to feature weighting is to have the weight be equal to the *distance* between the joint and the product distributions, as described in Eq. 2.16.

$$D_{KL}(P(X,C)||P(X)P(C)) = \sum_{c\in C, x\in X} P(c,x)\log\frac{P(c,x)}{P(x)P(c)} \tag{2.16}$$

Note that this is exactly the same as the mutual information between the feature and the class (see Eq. 2.9) so we have $D_{KL}(P(X,C)||P(X)P(C)) = MI(X,C)$.

### Correlation based methods

Even though this approach to feature weighting is treated last, maybe is one of the simplest as it does not care about continuous feature discretization or probability density estimations. It is usual in statistics to construct contingency tables for pairs of discrete variables to analyze their correlation. In our case (see Table 2.1) we will define a contingency table between the set of classes $c_i \in C$ and the values of a feature $x_j \in X$. The inner cells in row $i$ and column $j$ of the table contain the number of instances of class $c_i$ that have feature $X = x_j$. The row marginal totals will tell the number of instances for the corresponding class and the column marginal totals the number of instances with the corresponding value on feature $X$. Finally the sum of either marginal totals should be the total number of instances $m$. Looking at this

| | $x_1$ | $x_2$ | ... | $x_v$ | Tot. |
|---|---|---|---|---|---|
| $c_1$ | $N_{11}$ | $N_{12}$ | ... | $N_{1v}$ | $N_{1\cdot}$ |
| $\vdots$ | | | $\ddots$ | | $\vdots$ |
| $c_w$ | $N_{w1}$ | $N_{w2}$ | ... | $N_{wv}$ | $N_{w\cdot}$ |
| Tot. | $N_{\cdot 1}$ | $N_{\cdot 2}$ | ... | $N_{\cdot v}$ | $m$ |

| | |
|---|---|
| $v$ | No. of values for $X$ |
| $w$ | No. of classes ($C$) |
| $m$ | Total no. of instances |
| $N_{c_i\cdot}$ | Total no. in class $c$ |
| $N_{\cdot x_j}$ | Total no. with $X = x_j$ |
| $N_{c_i x_j}$ | No. with $C = c \wedge X = x_j$ |

**Table 2.1:** Contingency table of the class vs. the $X$ feature values

table we can define the Chi-squared weight for feature $X$ as shown in Eq. 2.17:

$$\chi^2(X) = \sum_{x\in X, c\in C} \frac{(N_{cx} - E_{cx})^2}{E_{cx}} \tag{2.17}$$

where $E_{cx}$ is the expected number of instances of class $c$ with value $x$ on feature $X$ calculated as $N_{c\cdot}N_{\cdot x}/m$. The value $\chi^2$ is distributed approximately as a $\chi^2$ random variable with $(v-1)(w-1)$ degrees of freedom. We should avoid terms with $E_{cx} = 0$ or replace them with a small positive number. We can see that in the extreme case that $X$ and $C$ are completely independent $N_{cx} = E_{cx}$ is expected so large values of $\chi^2(X)$ indicate strong dependence between the feature and the class. Note that the result of $\chi^2$ depends not only on the joint probabilities $P(c,x) = N_{cx}/m$ but also on the number of instances $m$. This latter dependency seems to make sense with the intuition that correlations calculated with small number of instances shall be less accurate.

## 2.2.1 Relief

One common characteristic of the previous methods is that they treat features individually assuming conditional independence of features upon the class. In contrast, Relief takes all

other features in care when evaluating a specific feature. Another interesting characteristic is that RELIEF is aware of contextual information, being able to detect local correlations of feature values and their ability to discriminate from an instance of a different class.

The main idea behind the alforithm is to assign large weights to features that contribute in separating near instances of different class and joining near instances belonging to the same class. The word "near" in the previous sentence is of crucial importance, since we mentioned that one of the main differences between RELIEF and other methods is the ability to take local context into account. RELIEF does not reward features that separate (join) instances of different (same) classes in general but features that do so for near instances.

**Algorithm 2.3:** Relief

**Input**: for each training instance a vector of feature values and the class value
**Output**: the vector W of estimations of the qualities of features

```
1 set all weights W[A] = 0.0;
2 for i = 1 to m do
3     randomly select an instance R_i;
4     find nearest hit H and nearest miss M;
5     for A = 1 to n do
6         W[A] := W[A] − diff(A, R_i, H)/m + diff(A, R_i, M)/m;
7     end
8 end
```

In **Algorithm 2.3** we can see the original algorithm as presented by Kira and Rendell [38]. We maintained the original notation that slightly differs from the used above as now features (attributes) are labeled $A$. There we can see that in the aim of detecting whether the feature is useful to discriminate near instances it selects two nearest neighbors of the current instance $R_i$. The one from the same class ($H$) is called the nearest hit and the one from a different class ($M$) is called the nearest miss[1]. With these two nearest neighbors it increases the weight of the feature if it has a similar value for both $R_i$ and $H$ and decreases it otherwise. The opposite occurs with the nearest miss: RELIEF increases the weight of a feature if it has dissimilar values for $R_i$ and $M$ and decreases it otherwise.

One of the central parts of Relief is the difference function diff, which is also used to compute the distance between instances as shown in Eq. 2.18.

$$\delta(\mathbf{x_1}, \mathbf{x_2}) = \sum_i \text{diff}(A_i, \mathbf{x_1}, \mathbf{x_2}) \tag{2.18}$$

The original definition of diff was an heterogeneous distance metric composed of the *overlap* metric in Eq. 2.19 for nominal features and the normalized Euclidean distance in Eq. 2.20 for linear features, which [83] called HEOM.

$$\text{diff}(A, \mathbf{x_1}, \mathbf{x_2}) = \begin{cases} 0 & \text{if value}(A, \mathbf{x_1}) = \text{value}(A, \mathbf{x_2}) \\ 1 & \text{otherwise} \end{cases} \tag{2.19}$$

[1] (The original RELIEF algorithm only dealt with two class problems).

$$\text{diff}(A, \mathbf{x_1}, \mathbf{x_2}) = \frac{|\text{value}(A, \mathbf{x_1}) - \text{value}(A, \mathbf{x_2})|}{\max(A) - \min(A)} \tag{2.20}$$

The normalization by $m$ guarantees that the weight range is $[-1, 1]$. In fact the algorithm tries to approximate a probability difference in Eq. 2.21.

$$\begin{aligned} W[A] \approx & P(\text{different value of } A | \text{nearest instance from different class}) - \\ & P(\text{different value of } A | \text{nearest instance from same class}) \end{aligned} \tag{2.21}$$

We can see that for a dataset $S$, where $|S| = d$ having a set of features $Y$, where $|Y| = n$, this algorithm has cost $O(m \times d \times n)$ as it has to loop over $m$ instances. For each instance in the main loop it has to compute its distance from all other instances so we have $O(m \times d)$ times the complexity of calculating $\delta$ and we can easily see from Eq. 2.18 that its complexity is $O(n)$, so we have our complexity: $O(m \times d \times n)$. As $m$ is a user defined parameter we can in some measure control the cost of Relief algorithm having a tradeoff between accuracy of estimation (for large $m$) and low complexity of the algorithm (for small $m$); however $m$ can never be greater than $n$.

### 2.2.2 Extensions of Relief

The first modification proposed to the algorithm is to make it deterministic by changing the outer loop through $m$ randomly chosen instances for a loop over all instances. This obviously increases the computational cost, which becomes $O(d^2 \times n)$ but makes experiments more reproducible –specially with small datasets. Kononenko uses this simplified version of the algorithm in his paper [43] to test his new extensions to the original RELIEF. This version is also used by other authors [41] and it is given the name RELIEVED with the final $d$ for "deterministic".

We can find some extensions to the original RELIEF algorithm proposed in [43] in order to overcome some of its limitations: it couldn't deal with incomplete datasets, it was very sensible to noisy data and it could only deal with multi-class problems by splitting the problem into series of two-class problems.

To enable RELIEF to deal with incomplete datasets, i.e. that contained missing values, a modification of the diff function is needed. The new function must be capable of calculating the difference between a value of a feature and a missing value and between two missing values in addition to the calculation of difference between two known values. Kononenko proposed various modifications of this function in its paper and found one that performed better than the others it was the one in a version of Relief he called RELIEF-D (not to be confused with the RELIEVED mentioned above). The difference function used by RELIEF-D can be seen in Eq. 2.22.

$$\text{diff}(A, \mathbf{x_1}, \mathbf{x_2}) = \begin{cases} 1 - P(value(A, \mathbf{x_2}) | class(\mathbf{x_1})) & \text{if } \mathbf{x_1} \text{ is missing} \\ 1 - \sum\limits_{a \in A} [P(a | class(\mathbf{x_1})) \times P(a | class(\mathbf{x_2}))] & \text{if both missing} \end{cases} \tag{2.22}$$

Giving RELIEF greater robustness against noise can be achieved by increasing the number of nearest hits and misses to look at. This mitigates the effect of choosing a neighbor that

would not have been the nearest without the effect of noise. The new algorithm has a new user defined parameter $k$ that controls the number of nearest neighbors to use. In choosing $k$ there is a tradeoff between locality and noise robustness –[43] states that $k = 10$ is a good choice for most purposes.

The last limitation was that the algorithm was only designed for two-class problems. The straightforward extension to multi-class problems would be to take as the near miss the nearest neighbor belonging to a different class. This variant of RELIEF was called RELIEF-E by Kononenko. However, later on, he proposes yet another variant which gave better results: this was to take the nearest neighbor (or the $k$ nearest) from each class and average their contribution so as to keep the contributions of hits and misses symmetric and between the interval $[0, 1]$. This produces the RELIEF-F (RELIEFF from now on) algorithm, seen in Fig. 2.2.

Input: for each training instance a vector of feature values and the class value
Output: the vector W of estimations of the qualities of features

1. set all weights $W[A] := 0.0$;
2. **for** $i := 1$ **to** $m$ **do begin**
3. randomly select an instance $R_i$;
4. find $k$ nearest hits $H_j$;
5. **for** each class $C \neq class(R_i)$ **do**
6. find $k$ nearest misses $M_j(C)$;
7. **for** $A := 1$ **to** $n$ **do**
8. $W[A] := W[A] - \sum_{j=1}^{k} \text{diff}(A, R_i, H_j)/(m \cdot k) +$
9. $\sum_{C \neq class(R_i)} \left[ \frac{P(C)}{1-P(class(R_i))} \sum_{j=1}^{k} \text{diff}(A, R_i, M_j(C)) \right] /(m \cdot k)$;
10. **end**;

**Figure 2.2:** Pseudo code of the RELIEFF algorithm

The aforementioned relation to impurity functions, in particular with the Gini-index gain in Eq. 2.8 can be seen in [70]. This is the case when developing the probability difference in Eq. 2.21 if the algorithm uses a large number of nearest neighbors (i.e., when the selected instance could be anyone from the set of instances). This version of the algorithm is called *myopic* ReliefF as it looses its context of locality property. Rewriting Eq. 2.21 by removing the neighboring condition and by applying Bayes' rule, we obtain Eq. 2.23.

$$W'[A] = \frac{P_{samecl|eqval} P_{eqval}}{P_{samecl}} - \frac{(1 - P_{samecl|eqval}) P_{eqval}}{1 - P_{samecl}} \tag{2.23}$$

For sampling with replacement we obtain we have:

$$P_{eqval} = \sum_{c \in C} P(c)^2$$

$$P_{samecl|eqval} = \sum_{x \in X} \left( \frac{P(x)^2}{\sum_{x \in X} P(x)^2} \times \sum_{c \in C} P(c|x)^2 \right)$$

Now we can rewrite Eq. 2.23 to obtain the myopic Relief weight estimation:

$$W'[A] = \frac{P_{eqval} \times GG'(X)}{P_{samecl} 1 - P_{samecl}} \tag{2.24}$$

Where $GG'(A)$ is a modified Gini-index gain of attribute $A$ as seen in Eq. 2.25.

$$GG'(X) = \sum_{x \in X} \left( \frac{P(x)^2}{\sum_{x \in X} P(x)^2} \times \sum_{c \in C} P(c|x)^2 \right) - \sum_{c \in C} P(c)^2 \tag{2.25}$$

As we can see the difference in this modified version from its original Gini-index gain described above in Eq. 2.8 is that Gini-index gain used a factor:

$$\frac{P(x)}{\sum_{x \in X} P(x)} = P(x)$$

while myopic ReliefF uses:

$$\frac{P(x)^2}{\sum_{x \in X} P(x)^2}$$

So we can see how this myopic ReliefF in Eq. 2.24 holds some kind of normalization for multi-valued attributes when using the factor $P_{eqval}$. This solves the bias of impurity functions towards attributes with multiple values. Another improvement compared with Gini-index is that Gini gain values decrease when the number of classes increase. The denominator of Eq. 2.24 avoids this abnormal behavior.

## 2.3 Stability

This section presents the state-of-the-art on stability of FSS methods. The subject has recently become a topic of interest –the first publication about FSS stability is from 2002 and the rest are between 2006 and 2008. We have found a more recent measure from Drotar and Smekal [16] in 2015 but we have not been able to reproduce their results so even though we explain it we are not using it in our experiments. The number of published papers is quite small (6) so the review in this section is exhaustive to our knowledge. This section is divided into three parts: one describing the exposed measures of stability, another describing the proposals on FSS results stability improvement and a final one exposing our review and conclusions on the first two parts. Every paper is not exposed as a whole but split to match this section's division. The aim of the first parts is not to make a critical review of the current publications but to expose them and establish a common notation and criteria. The limitations of the measures and improvements are discussed below on the last part 2.3.4.

### 2.3.1 Measures of stability

The above mentioned papers mainly focus on measuring stability of the feature selection methods, introducing measures based on Hamming distance, Dunne, Cunningham, and Azuaje [18], correlation coefficients and Jaccard index, Kalousis, Prados, and Hilario [36], consistency index, Kuncheva [45], Shannon entropy, Křížek, Kittler, and Hlaváč [44] and consistency measure Somol and Novovičová [73].

Let $Y = \{f_1, f_2, \ldots, f_n\}$ be the set of all features and let $\mathcal{X} = \{X_1, \ldots, X_m\}$ be a system of $m > 1 (m \in \mathbb{N})$ sets of feature subsets $X_j = \{f_i | i = 1, \ldots, d_j, f_i \in Y, d_j \in \{1, \ldots, n\}, j = 1, \ldots, m\}$ obtained from $m$ runs of a feature selection method. This set will also be referred as $\mathcal{X}_k$ when all its feature subsets have the same cardinality $k$.

#### Dunne, Cunningham, and Azuaje stability metric

Let $m_j = \{m_{j1}, \ldots, m_{jn}\}$ be a *feature mask*, a vector which indexes indicate the presence in $X_j$ of features from $Y$. Each element of the vector being defined as:

$$m_{ji} = \begin{cases} 1 & \text{if } f_i \in X_j, \\ 0 & \text{otherwise.} \end{cases}$$

Given a pair of feature masks, $m_i$ and $m_j$, we define the Hamming distance between them as follows:

$$D_H(m_i, m_j) = \sum_{k=1}^{n} |m_{ik} - m_{jk}| \tag{2.26}$$

To make this measure independent of the initial feature set length, we can normalize it dividing it by $n$. Also, the Hamming distance is a measure of *dissimilarity*, to convert to a similarity measure we can subtract it from 1. So, in set notation we have:

$$s_{Dunne}(X_i, X_j) = 1 - \frac{D_H(m_i, m_j)}{n} = 1 - \frac{|X_i \setminus X_j| + |X_j \setminus X_i|}{n} \tag{2.27}$$

This Hamming distance can be used to yield a measure of the overall variation of all the feature masks in $\mathcal{X}$. First we compute the total Hamming distance, $H_t$, by summing the individual Hamming distances between each pair of distinct masks:

$$s^t_{Dunne}(\mathcal{X}) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} s_{Dunne}(X_i, X_j) \tag{2.28}$$

In the above equation, $s^t_{Dunne}$, is computed over $P$ pairs of masks where $P$ is $m(m-1)/2$. By dividing by the number of pairs we have the stability metric $S_{Dunne}$ definition:

**Definition 2.5.** The Dunne, Cunningham, and Azuaje [18] stability metric $S_{Dunne}$ is a measure of stability based on the Hamming distance between the features masks in $\mathcal{X}$ defined as:

$$S_{Dunne}(\mathcal{X}) = \frac{2}{m(m-1)} s^t_{Dunne}(\mathcal{X}) \tag{2.29}$$

### Kalousis, Prados, and Hilario generalized similarity metric

Kalousis, Prados, and Hilario [36] introduced a stability measure between two feature sets $X_i$ and $X_j$, $J_i(X_i, X_j)$ as the Jaccard index between two sets:

$$J_i(X_i, X_j) = 1 - \frac{|X_i| + |X_j| - 2|X_i \cap X_j|}{|X_i| + |X_j| - |X_i \cap X_j|} = \frac{|X_i \cap X_j|}{|X_i \cup X_j|} \tag{2.30}$$

We present it here divided by $n$ to make it independent of the size of $Y$:

$$s_{Kalousis}(X_i, X_j) = \frac{1}{n} \frac{|X_i \cap X_j|}{|X_i \cup X_j|} \tag{2.31}$$

This similarity measure can again be used to yield a measure of the overall variation of all the feature sets in $\mathcal{X}$. The total similarity, $s^t_{Kalousis}$, by summing the individual similarities between each pair of sets:

$$s^t_{Kalousis}(\mathcal{X}) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S(X_i, X_j) \tag{2.32}$$

And normalizing with the number of terms in the sum we have the similarity metric definition:

**Definition 2.6.** The Kalousis, Prados, and Hilario generalized similarity metric $S_{Kalousis}$ is a stability measure based on an adaptation of the Jaccard index defined as:

$$S_{Kalousis}(\mathcal{X}) = \frac{2}{m(m-1)} s^t_{Kalousis}(\mathcal{X}) \tag{2.33}$$

### Kuncheva consistency index

Kuncheva [45] proposes a consistency index which is only applicable to subsets of the same cardinality $k$. In the paper some desired properties of the stability measures are pointed out. A comparison of the properties of the proposed consistency index and the previous measures is made showing that the previous measures don't match the desired properties. These properties are:

**Monotonicity** For a fixed subset size, $k$, and number of features, $n$, the larger the intersection between the subsets, the higher the value of the consistency index.

**Limits** The index should be bound by constants which do not depend on $n$ or $k$. The maximum value should be attained when the two subsets are identical.

**Correction for chance** The index should have a constant value for independently drawn subsets of features of the same cardinality, $k$.

The paper also shows that the two previous measures don't satisfy the correction for chance property described above. The proposed consistency index for two subsets $X_i$ and $X_j$ such that $|X_i| = |X_j| = k$, where $0 < k < n$ is:

$$s_{Kuncheva}(X_i, X_j) = \frac{|X_i \cap X_j| - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{|X_i \cap X_j| n - k^2}{k(n-k)} \tag{2.34}$$

Generalizing this index to the $m$ sets of features in $\mathcal{X}_k$, we can compute the total consistency of every pair of subsets as:

$$s^t_{Kuncheva}(\mathcal{X}_k) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} s_{Kuncheva}(X_i, X_j) \tag{2.35}$$

And normalizing with the number of terms in the sum we have:

**Definition 2.7.** The Kuncheva consistency index $S_{Kuncheva}$ is a stability measure based on the sizes of the union and intersection of subsets of the same cardinality defined as:

$$S_{Kuncheva}(\mathcal{X}_k) = \frac{2}{m(m-1)} s^t_{Kuncheva}(\mathcal{X}_k) \tag{2.36}$$

#### Křížek, Kittler, and Hlaváč stability measure based on Shannon's entropy

In their paper Křížek, Kittler, and Hlaváč [44] state that the previously proposed stability measures have many limitations, unclear motivation, and empirically estimated bounds. The authors suggest that the bounds of a stability measure should be reached in two extreme cases. The lower bound should be reached in the case of a random feature selections which selects every feature subset with the same probability and thus produces a uniform probability distribution. The upper bound would be reached by a feature selection method which all the time selected the same feature subset and thus creates a single peak probability distribution. So they suggest to assess the stability of feature selection methods based on the properties of the generated probability distributions of the selected feature subsets. The measure they propose to measure the randomness of these probability distributions is entropy [68]. In information theory, the concept of entropy indicates the amount of uncertainty about an event associated with a given probability distribution. The entropy is maximal for a uniform probability distribution (i.e., outcome of random feature selection). If the event is certain (i.e., outcome of perfectly stable feature selection) then the entropy is zero. So the desired properties are satisfied. Over the different entropy measures the authors derived their stability measure from Shannon's entropy [68]:

$$H(X) = -\sum_{i=1}^{m} P(x_i) \log P(x_i) \tag{2.37}$$

With $X$ being a discrete random variable with possible states $X = \{x_1, \ldots, x_m\}$, i.e. the particular feature subsets. $m \in \mathbb{N}$ is the number of all possible states, i.e. the number of possible different feature subsets, and $P(x_i)$ is the probability of the $i$-th state occurrence, i.e. the probability of selecting a particular feature subset. The authors only define the stability measure for subsets of a certain size $k$. So the probability of a certain subset $X_i$ are the number of occurrences of this subset in the $m$ subsets in $\mathcal{X}_k$ divided by all the possible subsets of $Y$ of size $k$ which is $C(n,k) = \binom{n}{k}$. So if the frequency of $X_i$ is $F_{X_i}$, the probability estimate of its occurrence can be determined by normalizing its occurrence by the number of subsets, i.e., $\overline{F}_{X_i} = F_{X_i}/m$. The proposed measure is then:

**Definition 2.8.** The Křížek, Kittler, and Hlaváč stability measure $S_{Krizek}$ for subsets in $\mathcal{X}_k$ of the same cardinality $k$ is a stability measure based on Shannon's entropy defined as:

$$S_{Krizek}(\mathcal{X}_k) = -\sum_{X_i \in \mathcal{X}_k} \overline{F}_{X_i} \log \overline{F}_{X_i} \tag{2.38}$$

**Somol and Novovičová consistency measures**

Somol and Novovičová [73] propose three novel stability measures and compare them to the generalized Kalousis, Prados, and Hilario generalized similarity metric $S_{Kalousis}$. The main difference between this method and the above mentioned ones is that all the above methods evaluate pairwise similarities between subsets in system $\mathcal{X}_k$ while the consistency measures evaluate the overall occurrence of features in the system as a whole.

Let $F_f$ be the number of occurrences (frequency) of feature $f$ in system $\mathcal{X}$ as defined below:

$$F_f = |\{X_i | X_i \in \mathcal{X}, f \in X_i\}| \tag{2.39}$$

Let $X$ be the subset of $Y$ representing all features that appear anywhere in $\mathcal{X}$. Let $N$ denote the number of all features in system $\mathcal{X}$, i.e., $N = \sum_{i=1}^{m} |X_i|$, so $N \in \mathbb{N}, N > n$. The stability measures proposed by the authors are:

**Definition 2.9.** The consistency $S_{Somol}(\mathcal{X})$ of system $\mathcal{X}$ is defined as

$$S_{Somol}(\mathcal{X}) = \frac{1}{|X|} \sum_{f \in X} \frac{F_f - 1}{m - 1} \tag{2.40}$$

**Definition 2.10.** The weighted consistency $S^W_{Somol}(\mathcal{X})$ of system $\mathcal{X}$ is defined as

$$S^W_{Somol}(\mathcal{X}) = \sum_{f \in Y} \frac{F_f}{N} \cdot \frac{F_f - 1}{m - 1} \tag{2.41}$$

Neither of these measures satisfies the correction for chance property stated by Kuncheva. The value of $S^W_{Somol}(\cdot)$ gets high when the sizes of the feature subsets approach the total number of features in $Y$ because in such system the subsets get necessarily more similar to each other. So the authors propose a last measure that do satisfy this property by normalizing $S^W_{Somol}(\cdot)$ by its range. The range of the function depend on the total number of features in the system $N$, the number of sets $m$ and the number of features $|Y| = n$. So for each of these values we can find lower and upper bounds for $S^W_{Somol}(\cdot)$, to be denoted $S^W_{min}(N, m, n)$ and $S^W_{max}(N, m)$ respectively.

**Definition 2.11.** The relative weighted consistency $S^{rel}_{Somol}(\mathcal{X})$ of system $\mathcal{X}$ and for given $Y$ is defined as

$$S^{rel}_{Somol}(\mathcal{X}) = \frac{S^W_{Somol}(\mathcal{X}) - S^W_{min}(N, m, n)}{S^W_{max}(N, m) - S^W_{min}(N, m, n)} \tag{2.42}$$

No details on how to compute $S^W_{min}(N, m, n)$ and $S^W_{max}(N, m)$ will be given here for the sake of simplicity, they can be found at [73].

### Drotar and Smekal histogram stability measure

Drotar and Smekal [16] introduce a novel stability measures and compare it to some previous existing measures described above. Their measure which we will call $S_{Drotar}$, express the stability as the ratio between average number of feature occurrences of $T$ features with highest occurrence and average number of feature occurrences of other features. Let $F_{top} \subset F$ contain $T$ features with highest occurrence $N_f$ and $F_{other} = F \setminus F_{top} \setminus F_0$ contain all other features with $N_f \neq 0$. Here, $F_0 = \{f_1, \ldots, f_q\}$, where occurrences $N_{f_1} = \cdots = N_{f_q} = 0$.

$$S_{Drotar}(\mathcal{X}) = \frac{\frac{1}{|F_{top}|} \sum_{f \in F_{top}} N_f}{1 + \frac{1}{|F_{other}|} \sum_{f \in F_{other}} N_f} \tag{2.43}$$

## 2.3.2 Experimental results

The aim of this section is to test the above described theoretical properties of the measures in some datasets to show the practical effect of their differences. We analyze behavior of FSS stability measures through two different perspectives: Influence of randomness in FSS process and subset size.

### Results on artificial data

The first evaluation will be with the synthetic data proposed by Kuncheva [45]. Assume scenario where number of all features $n = 10$ and $m = 2$ runs of FS algorithm were performed to obtain subset of selected features. **Figure 2.3** shows the values of stability measures as the function of subset size $k$. First $k$ features of $X_1$ and $X_2$ are included in subset of selected features. The $X_1$ and $X_2$ are as follows:

$$X_1 = \{f_9, f_7, f_2, f_1, f_3, f_{10}, f_8, f_4, f_5, f_6\} \tag{2.44}$$

$$X_2 = \{f_3, f_7, f_9, f_{10}, f_2, f_4, f_8, f_6, f_1, f_5\} \tag{2.45}$$

We have reproduced the original formula for $S_{Drotar}$ as described in [16] but we have not been able to reproduce their results. In fact, is easy to see that with the given formula the results will be constant for this problem as $F_{top}$ is always 2 and $F_{other}$ is always 1 for $k \geq 2$. For $k = 1$ the above formula is not well defined as the value would be $0/0$. But the authors show varying results so there must be an error with the formula above. For this reason we have decided to exclude $S_{Drotar}$ from our results.

As we can see all stability measures correctly identified a decrease in stability at $k = 4$. But we can note that there is different behavior of stability measures with increasing subset size for $k > 5$. $S_{Dunne}$ is almost constant even though the feature choices vary. $S_{Kalousis}$ does not have a correction by chance so as more and more features are selected its value increases even though the real robustness is not greater. As $S_{Krizek}$ only measures frequencies of subsets of features and the two sequences start with a different feature, theres is no single subset repetition leading to a constant value of 0. $S^W_{Somol}$ and $S_{Kuncheva}$ behave quite simialar and are the only measures that are sensible enough to detect the stability changes and have the correction by chance to detect that the stability decreases at the end.

In the next experiment we are maximizing the effect of noisy choices to illustrate even more the weaknesses of some of the measures that have no correction by chance. Similarly

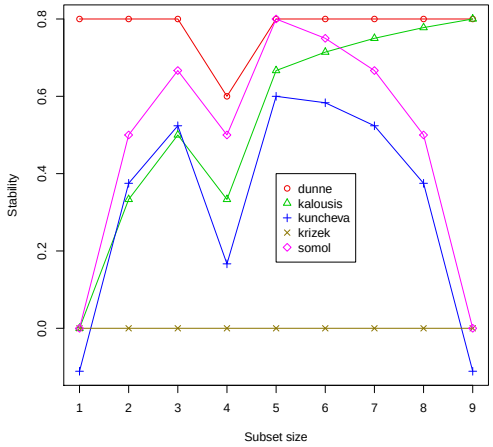



**Figure 2.3:** Comparison of stability measures for sequences $X_1$ and $X_2$ as function of subset size $k$

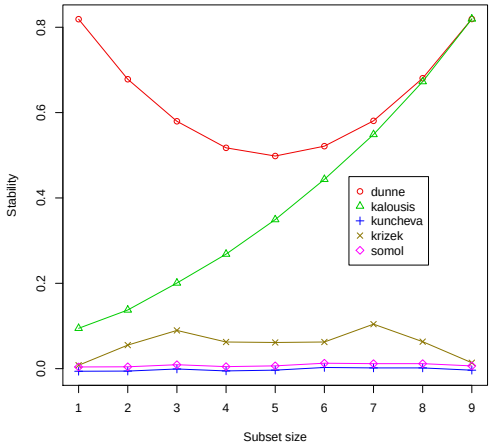



**Figure 2.4:** Comparison of stability measures for 100 randomly generated sequences as function of subset size $k$

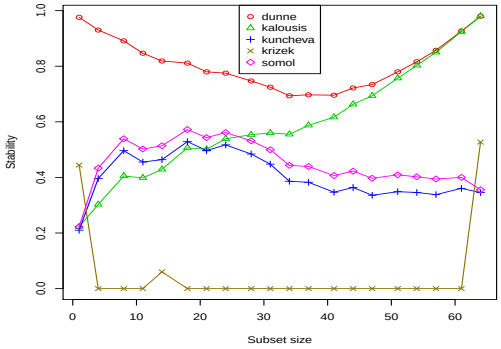
to what Kuncheva [45] also proposes, we generate 100 independent random sequences of 10 features. Again we will build subsets of features by selecting the first $k$ features will study the effect of increasing $k$. As the resulting feature subsets are completely random, a robust stability measure should give us values that are close to 0. Again only $S^W_{Somol}$ and $S_{Kuncheva}$ have the desired behavior as shown in **Figure 2.4**.

### Results on real data

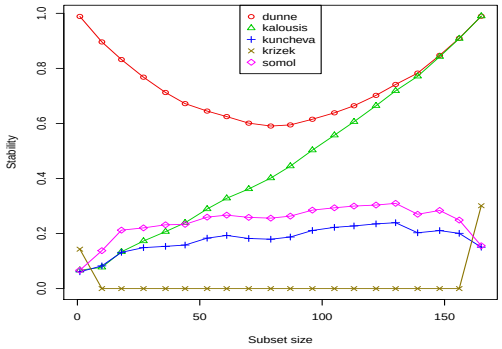
To validate the results also on real databases we evaluatefeature selection stability on various datasets as described in Section 3.1.As a feature selection method we had have used both ReliefF and Simba, the two filter algorithms that we deeply analyze in this thesis. In this case we have chosen two *filter* algorithms for thier low computational time as we were not pursuing to find the best possible subset but to evaluate the behavior of the different stability measures on the results. Figures 2.5 to 2.8 show the results for all the different dataset and agorithm combinations. Again we can see the same behavior as with the artificial datasets. $S^W_{Somol}$ and $S_{Kuncheva}$ provide very similar results. These results indicate that both measures are concise and are robust to changes in subset size. Dunne stability measure $S_{Dunne}$ shows unsatisfactory behavior, providing values close to 1 for large datasets. $S_{Krizek}$ fails to detect the similarities and provide values close to 0 for every dataset and algorithm and $S_{Kalousis}$ is highly influenced by high dimensionality of dataset.

### Conclusions on the stability measures

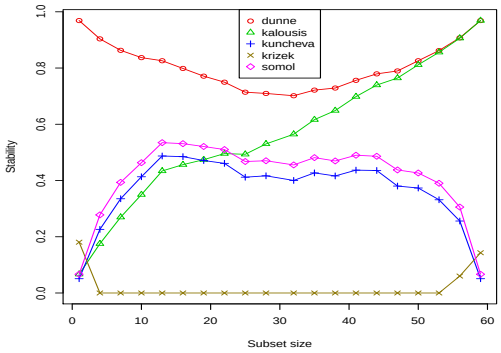
Only two of the presented stability measures give consistent results and have the needed correction by chance: $S^W_{Somol}$ and $S_{Kuncheva}$. In addition the results provided by the two



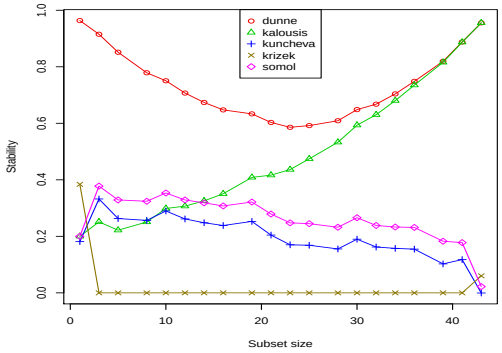

a Mammogram



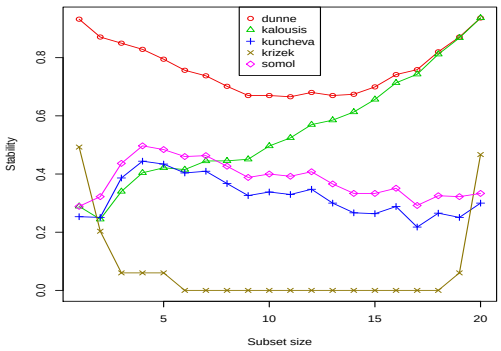

b Musk



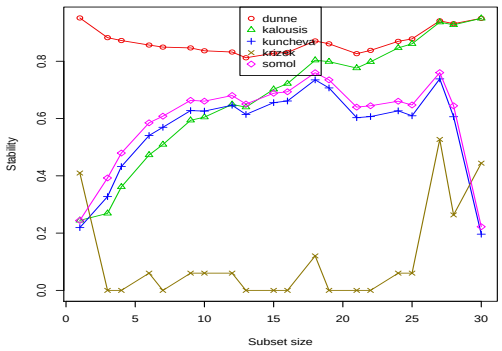

c Sonar




d SpectF

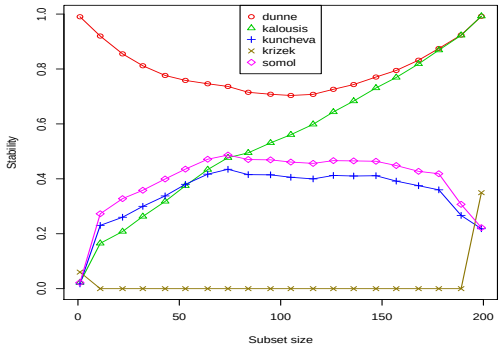



e Waveform

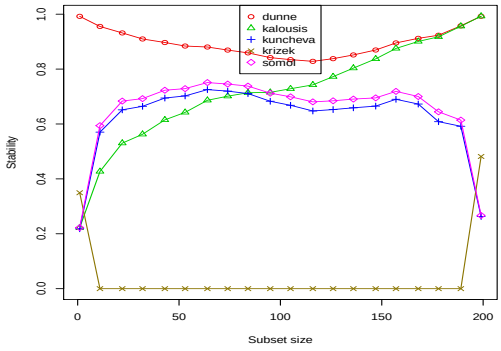



f WDBC

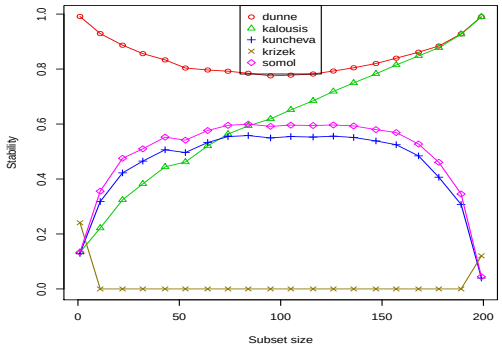
**Figure 2.5:** Stability measures for ReliefF on UCI datasets



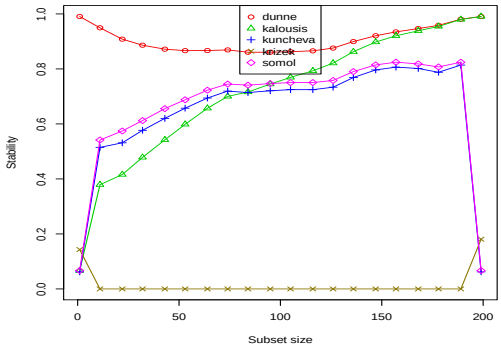

a Breast cancer



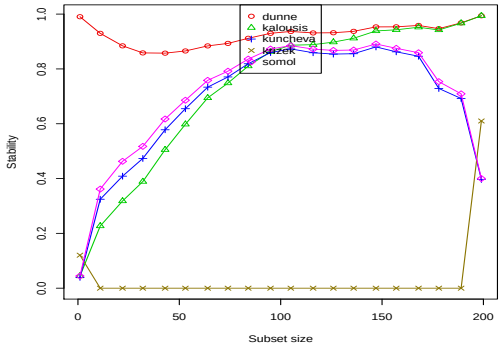

b Musk



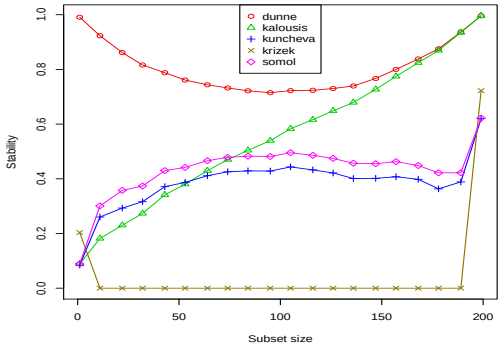

c GCM




d Leukemia

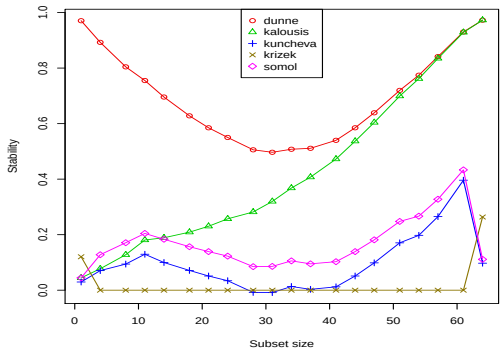



e Lung cancer

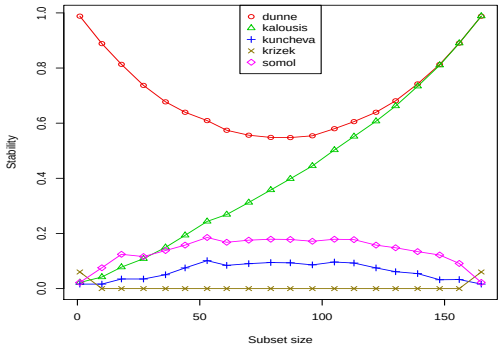



f Prostate cancer

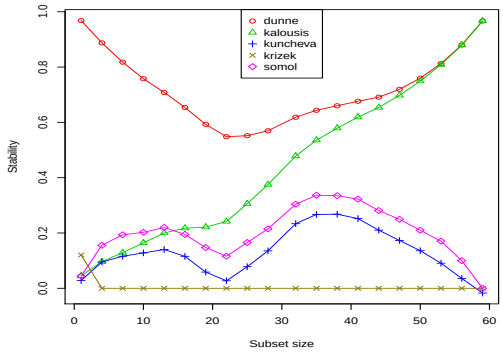
**Figure 2.6:** Stability measures for ReliefF on microarray datasets



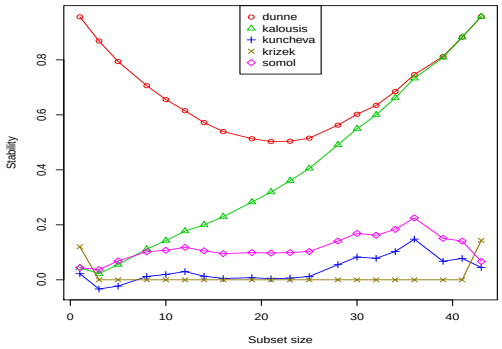

a Mammogram



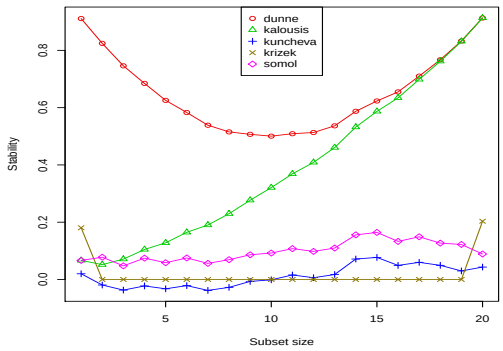

b Musk



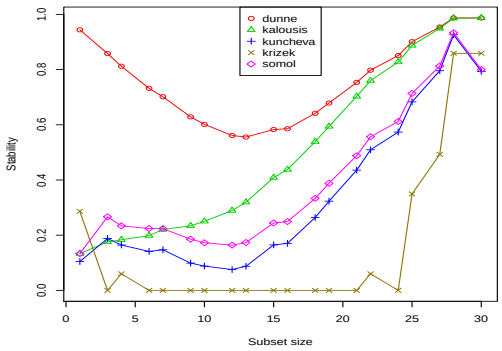

c Sonar




d SpectF

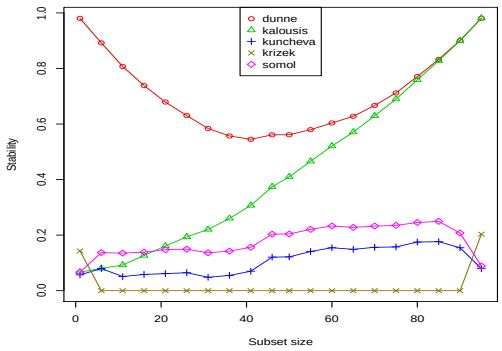



e Waveform

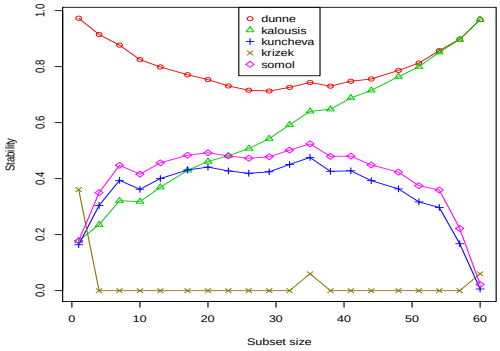



f WDBC

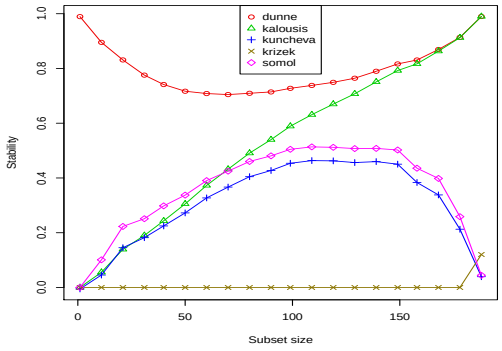
**Figure 2.7:** Stability measures for SIMBA on UCI datasets



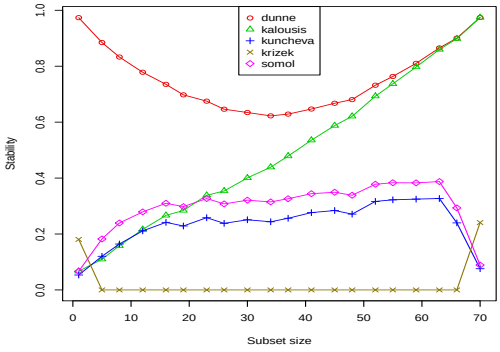

a Breast cancer



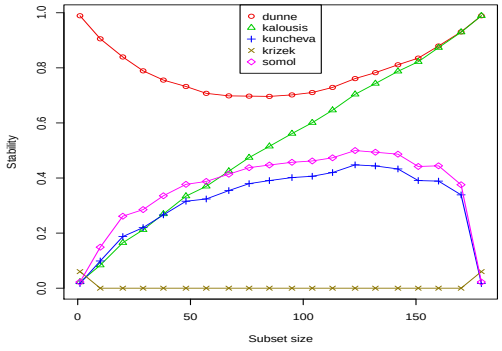

b Musk



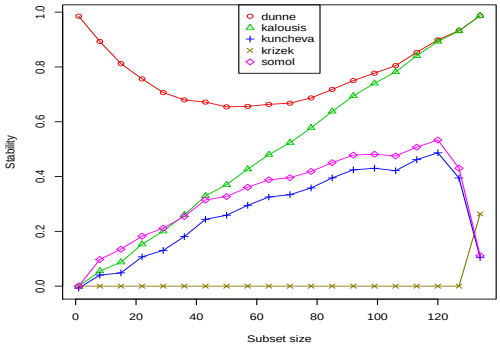

c GCM




d Leukemia




e Lung cancer




f Prostate cancer

**Figure 2.8:** Stability measures for SIMBA on microarray datasets

measures are very similar, even identical in some cases. For these reasons we will only be using [45] stability measure $S_{Kuncheva}$ in further experiments throughout this work.

### 2.3.3 Improvements of stability

Here we present the current proposed improvements on stability found in literature. Different approaches have been taken in order to improve stability, below we list each of them, give references to the original sources and make a brief description of the proposal.

#### Multiple runs of the FS algorithm

Dunne, Cunningham, and Azuaje [18] propose a method to increase the stability of wrapper feature selection methods solutions based on executing the algorithm multiple times and selecting the features that appeared the most in the system of solutions. The authors see this as applying a wrapper to the wrapper, so they call it the Wrapper-2 approach. Using the notation above, $\mathcal{X}$ is the system of subsets of features resulting from a feature selection process over $Y$. $F_f$ is the number of occurrences (frequency) of feature $f$ in the system, as defined in (2.39). The Wrapper-2 idea is to select the features with higher values of $F_f$. The paper mentions two possible selection criteria: Selecting the $k$ features $\{f_1, \ldots, f_k\}$ with highest values of $F_{f_i}$ or selecting features incrementally in rank order (as indicated by their $F_{f_i}$ value) and evaluate the feature set by executing the inducer on a holdout set of examples until adding more features no longer increased the performance. In the paper two experiments are conducted. One to compare the stability of $k$ runs of the aggregated method and $k$ runs of the normal method for three different wrapper feature selection search strategies: SFG, SBG and RHC (Random Hill-Climbing search). And another to compare the performance of the aggregated method with the original one. Ensemble techniques are said to have better stability than single classifiers although no significance tests are provided. The performance of the aggregated methods is also superior again with no significance tests run.

The authors also suggest that increasing the coverage of the search strategy may lead to more stable feature selection methods but state that its evaluation was still at an early stage.

#### Re-sampling of the criterion estimate

Křížek, Kittler, and Hlaváč [44] state that the key factor for improving stability is improving the estimate $\mathcal{L}(\cdot)$ of the objective function values. The reason is that if the criterion is better, the search algorithm is more likely to find the optimal solution giving more stable results. Their improvement on the evaluation criterion estimate is based on re-sampling techniques such as cross-validation, holdout validation or bootstrap. They present the results of an experiment comparing the stability of a wrapper and a filter of the SFFS [57] feature selection methods using various n-fold cross-validations, holdout validations and various bootstrap variations. The filter version applies the Mahalanobis distance in the objective function definition. The wrapper form uses prediction accuracy of a linear decision rule created by the Gaussian classifier. Their results showed that the stability of the feature selection results increased when using re-sampling techniques for the evaluation of the $\mathcal{L}(\cdot)$ criterion. They found that for the filter, The filter variant of the SFFS algorithm achieves better stability if more samples are employed in the objective function estimation, i.e., using techniques like ten-fold or leave-one-out cross-validation, for instance. On the other hand, the wrapper achieved

the best stability with the 50/50 holdout validation. The authors argue that wrappers appear to be much more sensitive to the correct objective function estimate than filters. So when less data is used for the validation, the estimation's variance increases and so the algorithm becomes more sensitive to random perturbations in the data and fails to find a consistent solution. The .632 bootstrap achieved the best stability factor, however, its performance was by far the worst. Bootstrap techniques are supposed to give estimates with low variance [19] which explains a good stability. Nevertheless, the bias of the estimate is high and as a result the wrapper converged to a wrong solution.

#### Ensemble feature selection methods

Saeys, Abeel, and Peer [65] conduct a study of the stability of ensemble feature selection techniques using the Jaccard index as the similarity measure the same as the used in [36]. The hypothesis is that similarly to the case of supervised learning, ensemble techniques might be used to improve the stability of feature selection techniques. Indeed, in domains with many features and few examples, it is often reported that several different feature subsets may yield equally optimal results, and ensemble feature selection may reduce the in the paper is done by adding up the feature rankings provided by the single feature selectors into a final consensus ranking. Four feature selection methods were tested all being filter or embedded comparing the stability of the single version with the ensemble one. For each of the feature selection techniques, an ensemble version was created by instance perturbation using bagging [6] to generate 40 bags from the data. For each of the bags, a separate feature ranking was performed, and the ensemble was formed by aggregating the single rankings as mentioned above. The results showed that the stability of the feature selection methods increased when the number of bags increased while the performance of the algorithm was the same or slightly better.

### 2.3.4 Conclusions on the stability state-of-the-art

In an effort to simplify the notation and unify some of the proposed methods we will introduce some notation for the *similarity* measures, in which the presence of the feature is denoted by $+$ and its absence by $-$. For any two feature subsets $X_i, X_j$ of the set $Y$, to be compared on the basis of a feature $k$, a score $s_{ijk}$ can be defined, described below. First $\delta_{ijk}$ is defined as 0 when the comparison of $X_i, X_j$ cannot be performed on the basis of feature $k$ for some reason (e.g., by because we are not willing to count a $-,-$ match as a real match); $\delta_{ijk}$ is 1 when such comparison is meaningful. The coefficient of similarity between $X_i, X_j$ is defined as the average score over all the partial comparisons.

$$S_{ij} = \frac{\sum_{k=1}^{n} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{n} \delta_{ijk}}. \tag{2.46}$$

With this formulation we can obtain multiple similarity scores by assigning different values to $\delta_{ijk}$ and $s_{ijk}$ depending on what we consider to be a match. For instance if we assigned

the values as in the following table:

| | Values | | | |
|---|---|---|---|---|
| $X_i$ | + | + | − | − |
| $X_j$ | + | − | + | − |
| $s_{ijk}$ | 1 | 0 | 0 | 0 |
| $\delta_{ijk}$ | 1 | 1 | 1 | 0 |

we would obtain Jaccard index based similarity score like the one used in [36].

Then we call $a$ the number of $+,+$ matches, $b$ and $c$ the number of $+,-$ and $-,+$ matches and $d$ the number of $-,-$ matches. We can see a graphical representation of the sets in the following Venn diagram in figure 2.9.

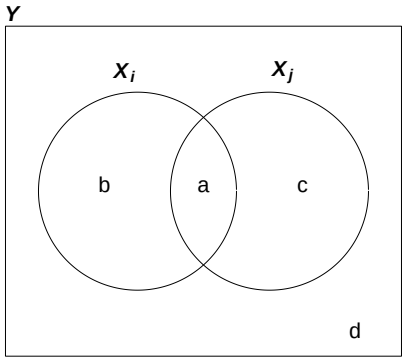



**Figure 2.9:** Venn diagram showing the $a, b, c, d$ values.

We will now rewrite the similarity based measures using the $a, b, c, d$ values and add the other two measures just to have a compact summary of all measures. Let us first recall some notation: $\overline{F}_{X_i}$ is the probability estimate of the occurrence of $X_i$ in $\mathcal{X}_k$, i.e. the number of occurrences of $X_i$ , divided by the total number of subsets, $m$. $F_f$ the frequency of feature $f$ in $\mathcal{X}$ and $N > n$ the number of all features in $\mathcal{X}$ (with repetitions). Finally, that both $\mathcal{X}$ and $\mathcal{X}_k$ represent a system of $m$ subsets of $Y$ with the only difference that the size of the subsets in the latter is fixed to $k$ as $S_{Krizek}(\cdot)$ and $S_{Kuncheva}(\cdot)$ are only defined for systems

of subsets of this same size $k$.

$$S_{Dunne}(\mathcal{X}) = \frac{2}{m(m-1)} \sum_{X_i \in \mathcal{X}} \sum_{X_j \in \mathcal{X} \setminus X_i} \frac{a+d}{n} \tag{2.47}$$

$$S_{Kalousis}(\mathcal{X}) = \frac{2}{m(m-1)} \sum_{X_i \in \mathcal{X}} \sum_{X_j \in \mathcal{X} \setminus X_i} \frac{a}{n-d} \tag{2.48}$$

$$S_{Kuncheva}(\mathcal{X}) = \frac{2}{m(m-1)} \sum_{X_i \in \mathcal{X}} \sum_{X_j \in \mathcal{X} \setminus X_i} \frac{an-k^2}{k(n-k)} \tag{2.49}$$

$$S_{Krizek}(\mathcal{X}_k) = -\sum_{X_i \in \mathcal{X}_k} \overline{F}_{X_i} \log \overline{F}_{X_i} \tag{2.50}$$

$$S^W_{Somol}(\mathcal{X}) = \sum_{f \in Y} \frac{F_f}{N} \frac{F_f - 1}{m-1} \tag{2.51}$$

$$S^{rel}_{Somol}(\mathcal{X}) = \frac{S^W_{Somol}(\mathcal{X}) - S^W_{min}(N,m,n)}{S^W_{max}(N,m) - S^W_{min}(N,m,n)} \tag{2.52}$$

As Kuncheva [45] state, a good similarity measure should be: monotone, bounded and have some correction by chance (See section 2.3.1 for more details). They also show that the $S_{Dunne}$ and $S_{Kalousis}$ measures are not corrected by chance so both tend to increase when the size of the selected set approaches the total number of features $n$, this being a serious limitation for these two measures. Another point of view is that of Křížek in his doctoral thesis. There, strong concerns on the motivation and the empirically estimated bounds of $S_{Dunne}$, $S_{Kalousis}$ and $S_{Kuncheva}$ are exposed.

Finally, the only measure that has been further used and analyzed is the one proposed in $S_{Kalousis}$, which is compared to $S^W_{Somol}$ and $S^{rel}_{Somol}$ in [73] and used in the experiments of [65] even though the papers describing the other previous measures are cited in both papers. Authors do not clarify the reasons why they do not use the other measures for their experiments and comparisons.

A strong weakness of entropy-based measures of stability that only focus on the distribution of feature subsets (like [44]) is that they only consider equality or non-equality of the obtained feature subsets, disregarding the important information present in the features that match or mismatch. To make this point clear, consider the following scenarios, in which we have five possible outcomes (solution feature subsets) from a selection process carried out in an initial set of size 20:

```
11110000000000000000        11111110000111111111
00001111000000000000        11111101000111111111
00000000111100000000        11111100100111111111
00000000000011110000        11111100010111111111
00000000000000001111        11111100001111111111
```

It should be clear that the situation on the left panel is a nightmare from the point of view of feature selection and its stability. In contrast, the situation on the right panel is much

better: the five selections only differ in two features when compared one against another. However, the entropy of both distributions is the same.

Another weakness shared among all the measures, is that even though all of the papers use SFSAs, no measure takes some important information about this particular search strategy into account. They ignore the fact that features are added in order into the feature subsets. However, this order has much to do with stability: among other reasons, because the selection of the first features greatly conditions the selection of the subsequent features and thus greatly influences the final selected subset. So maybe the choice of a different feature in a step where every feature can be selected (e.g. at the beginning of the SFG) should have less importance in stability calculation than differences when very few choices are available (e.g. at the end of the SFG).

Moreover, as [44] explicitly states, *stability does not say anything about the performance* of the selected features. Indeed, none of the proposed measures of stability takes performance into account, only [65] introduces a method to balance stability against classification performance by using an adaptation of the F-measure [78]. But none of them clarifies how stability should influence the feature selection process.

Reviewing the proposed improvements, we observe that they are all based on the idea of multiple runs of the feature evaluation criterion, either by running the whole feature selection process multiple times [18], by resampling the data on every feature evaluation in every step of the selection [44] or by running an ensemble of feature selectors [65]. So, all of the proposed improvements are only to deal with the problem of data resampling. Besides, none of the comparisons carried out in these papers give statistical significance of the stability differences between the original algorithm and the proposed improvement.

Finally, an almost neglected idea in literature is that stability could be used to somehow guide the feature selection process. Only Kalousis, Prados, and Hilario [36] points out as future work that stability can provide an objective criterion on which the feature choices can be based during feature selection in the absence of any significant difference in classification performance. It seems clear to us that if we wanted to improve the stability of a feature selection algorithm it should be taken into account during the process.

## 2.4 Instance margins

In machine learning the margin of an instance with respect to a classification rule measure the classifier confidence when making its decision. [13] describes two different approaches to define the margin of a particular instance $\mathbf{x}$ to a set of points $S$.

**Definition 2.12.** The *sample margin* is the distance between the instance and the decision boundary induced by the classifier. Support Vector Machines [12] are based on this definition of margin. See Figure 2.10a.

**Definition 2.13.** The *hypothesis margin* is the distance that the classifier can travel without changing the way it labels the instance. Note that this definition requires a distance measure between classifiers. This type of margin is used in AdaBoost [20]. See Figure 2.10b.

For 1-NN, the classifier is defined by a set of training points and the decision boundary is the Voronoi tessellation (See Figure 2.11). The sample margin in this case is the distance

**(a)** Sample Margin **(b)** Hypothesis Margin

**Figure 2.10:** Comparison of the two types of margins described above.

between the instance and the Voronoi tessellation, and therefore it measures the sensitivity to small changes of the instance position. The margins for 1-NN were described by [13] and the following results were proved:

1. The hypothesis margin lower bounds the sample margin
2. It is easy to compute the hypothesis margin of an instance $\mathbf{x}$ to a set o points $S$ by the following formula:

$$\theta_S(\mathbf{x}) = \frac{1}{2}\left(|\mathbf{x} - \mathbf{nearmiss}(\mathbf{x})| - |\mathbf{x} - \mathbf{nearhit}(\mathbf{x})|\right) \tag{2.53}$$

were $\mathbf{nearhit}(\mathbf{x})$ and $\mathbf{nearmiss}(\mathbf{x})$ are the nearest points to $\mathbf{x}$ in $S$ with the same class and with a different class respectively.

### 2.4.1 Margin based instance weighting (MBIW)

Han and Yu [29] describe a framework of instance weighting employing the concept of margins in a different way of the used in the large margin principles [26], [3]. By decomposing the margin of an instance along each dimension, the instance in the original feature space can be represented by a new vector (called margin vector) in the margin vector feature space defined as follows.

**Definition 2.14.** Let $\mathbf{x} = (x_1, \ldots, x_d)$ be an instance in the original feature space $\mathfrak{R}_d$, and $\mathbf{nearhit}(\mathbf{x})$ and $\mathbf{nearmiss}(\mathbf{x})$ represent the nearest instances to $\mathbf{x}$ with the same and different class, respectively. Every instance $\mathbf{x} \in \mathfrak{R}_d$ can be mapped to $\mathbf{x}'$ according to:

$$x'_j = |x_j - \mathbf{nearmiss}(\mathbf{x})_j| - |x_j - \mathbf{nearhit}(\mathbf{x})_j| \tag{2.54}$$

where $x'_j$ is the $j$th coordinate of $\mathbf{x}'$ in the new feature space $\mathfrak{R}'_d$, and $x_j$, $\mathbf{nearhit}(\mathbf{x})_j$, or $\mathbf{nearmiss}(\mathbf{x})_j$ is the $j$th coordinate of $\mathbf{x}$, $\mathbf{nearhit}(\mathbf{x})$ or $\mathbf{nearmiss}(\mathbf{x})$ in $\mathfrak{R}_d$, respectively. Vector $\mathbf{x}'$ is called the *margin vector* of $\mathbf{x}$, and $\mathfrak{R}'_d$ is called the *margin vector feature space*.

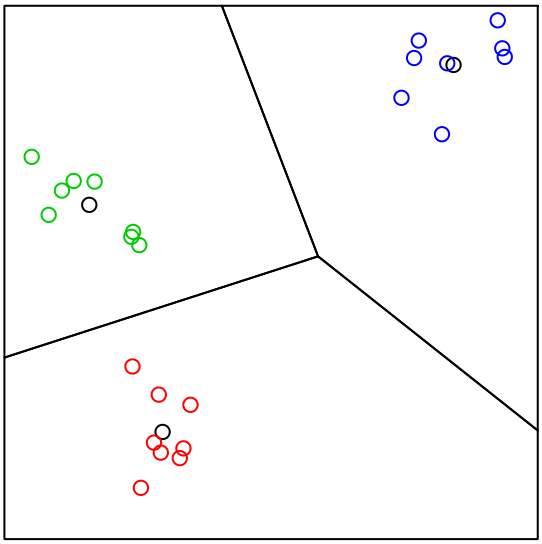


**Figure 2.11:** Voronoi tessellation for the 1-NN classifier

In order to reduce the effect of noise and outliers in the training set, the authors do not use one nearest neighbor as described in Eq. (2.54) but all the neighbors from each class. So the margin vector definition can be extended as:

$$x'_j = \sum_{l=1}^{m} |x_j - \mathbf{nearmiss}(l, \mathbf{x})_j| - \sum_{l=1}^{h} |x_j - \mathbf{nearhit}(l, \mathbf{x})_j| \tag{2.55}$$

where $\mathbf{nearhit}(l, \mathbf{x})_j$ and $\mathbf{nearmiss}(l, \mathbf{x})_j$ is the $j$th component of the $l$th neighbor to $\mathbf{x}$ of the same class and of different class, respectively. $h$ represents the total number of instances with the same class in the training set (i.e. hits) and $m$ the total number of instances of different class (i.e. misses). Note that $h + m$ is the total number of instances in the training set excluding $\mathbf{x}$.

With this definition we see that the larger the value of $x'_j$, the more the feature $j$ contributes to the margin of instance $\mathbf{x}$. $\mathbf{x}'$ captures the local profile of feature relevance for all features at $\mathbf{x}$. Then to compute overall relevance for each feature, one idea is to take average over all margin vectors as relief does.

However the authors suggest not to take the average of all instances but to weight their contribution based on their projections into the margin space. They state that more stable feature weightings can be obtained by reducing the influence of instances that exhibit distinct margin vectors from the majority of the instances as the presence or absence of these instances will highly affect the decision on which feature is more relevant. Specifically the weight of an

instance $\mathbf{x}$ is given by:

$$w(\mathbf{x}) = \frac{1/\overline{dist}(\mathbf{x}')}{\sum_{i=1}^{n} 1/\overline{dist}(\mathbf{x}'_i)} \tag{2.56}$$

where

$$\overline{dist}(\mathbf{x}') = \frac{1}{n-1} \sum_{i=1, \mathbf{x}'_i \neq \mathbf{x}'}^{n-1} dist(\mathbf{x}', \mathbf{x}'_i)$$

**Algorithm 2.4** describes the **MBIW** process for assigning weights to instances using the definitions above.

**Algorithm 2.4:** Margin Based Instance Weighting (MBIW)

**Input**: training data $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i = \{x_{i,1}, \ldots, x_{i,d}\}$
**Output**: weight vector $\mathbf{w} = \{w_1, \ldots, w_n\}$ for all instances in $S$
`// Feature Space Transformation`
**for** $i = 1$ **to** $n$ **do**
  **for** $i = j$ **to** $d$ **do**
    $x'_{i,j} = \sum_{l=1}^{m} |x_j - \mathbf{nearmiss}(l, \mathbf{x})_j| - \sum_{l=1}^{h} |x_j - \mathbf{nearhit}(l, \mathbf{x})_j|$
  **end**
**end**
`// Instance Weighting`
**for** $i = 1$ **to** $n$ **do**
  $w_i = \frac{1/\overline{dist}(\mathbf{x}'_i)}{\sum_{j=1}^{n} 1/\overline{dist}(\mathbf{x}'_j)}$
**end**

Once weights have been assigned to features a regular feature selection algorithm that can take instance weights into account can be run to find relevant features. So the instance weighting is presented by the authors as a *preprocessing* step before applying one of the current feature selection methods.

### 2.4.2 Margin based feature selection (Simba)

Bachrach, Navot, and Tishby [3] present a novel feature selection algorithm c that gives weights to features based on their contributions to instances' margins. The main idea is that a good generalization can be guaranteed if many sample points have large margins so one should select features that contribute more to these margins.

**Definition 2.15.** Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the training set of instances and $\mathbf{x} = \{x_1, \ldots, x_d\}$ be a particular instance in $S$. Let $\mathbf{w} = \{w_1, \cdots, w_d\}$ be a weight vector over the feature set, then the *margin* of $\mathbf{x}$ is

$$\theta_S^{\mathbf{w}}(\mathbf{x}) = \frac{1}{2} \left( \|\mathbf{w} - \mathbf{nearmis}(\mathbf{x})\|_{\mathbf{w}} - \|\mathbf{w} - \mathbf{nearhit}(\mathbf{x})\|_{\mathbf{w}} \right) \tag{2.57}$$

where $\|\mathbf{z}\|_{\mathbf{w}} = \sqrt{\sum_i w_i^2 z_i^2}$

**Definition 2.16.** Let $u(\cdot)$ be a utility function. Given the training set $S$ and the weight vector $\mathbf{w}$, the *evaluation function* is:

$$e(\mathbf{w}) = \sum_{\mathbf{x} \in S} u\left(\theta^{\mathbf{w}}_{S \setminus \mathbf{x}}(\mathbf{x})\right) \tag{2.58}$$

The utility function controls the contribution of each margin term to the overall score. It is natural to require the utility function to be non-decreasing; thus larger margin introduce larger utility. We consider three utility functions: linear, zero-one and sigmoid. The linear utility function is defined as $u(\theta) = \theta$. When the linear utility function is used, the evaluation function is simply the sum of the margins. The zero-one utility is equals 1 when the margin is positive and 0 otherwise. When this utility function is used the utility function is proportional to the leave-one-out error. The sigmoid utility is $u(\theta) = 1/(1+exp(-\beta\theta))$. The sigmoid utility function is less sensitive to outliers than the linear utility, but does not ignore the magnitude of the margin completely as the zero-one utility does. Note also that for $\beta \to 0$ or $\beta \to \infty$ the sigmoid utility function becomes the linear utility function or the zero-one utility function respectively. In the SIMBA algorithm we assume that the utility function is differentiable, and therefore the zero-one utility cannot be used.

It is natural to look at the evaluation function solely for weight vectors $\mathbf{w}$ such that $\max w_i^2 = 1$. However, formally, the evaluation function is well defined for any $\mathbf{w}$, a fact which we make use of in the SIMBA algorithm. We also use the notation $e(F)$, where $F$ is a set of features to denote $e(\mathbf{x_F})$.

The gradient of $e(\mathbf{w})$ when evaluated on a sample $S$ is:

$$\begin{aligned}
(\bigtriangledown e(\mathbf{w}))_i &= \frac{\partial e(\mathbf{w})}{\partial w_i} = \sum_{\mathbf{x} \in S} \frac{\partial u(\theta(\mathbf{x}))}{\partial \theta(\mathbf{x})} \frac{\partial \theta(\mathbf{x})}{\partial \mathbf{w}_i} \\
&= \frac{1}{2} \sum_{\mathbf{x} \in S} \frac{\partial u(\theta(\mathbf{x}))}{\partial \theta(\mathbf{x})} \Big( \frac{(x_i - \mathbf{nearmiss}(\mathbf{x})_i)^2}{\|\mathbf{x} - \mathbf{nearmiss}(\mathbf{x})\|_{\mathbf{w}}} - \frac{(x_i - \mathbf{nearhit}(\mathbf{x})_i)^2}{\|\mathbf{x} - \mathbf{nearhit}(\mathbf{x})\|_{\mathbf{w}}} \Big) w_i
\end{aligned}$$

In **Algorithm 2.5** we can see SIMBA using a stochastic gradient ascent over $e(w)$ while ignoring the constraint $\max w_i^2 = 1$, the projection on the constraint is only done at the end

(See line 10). This is sound since $e(\lambda w) = \lambda e(w)$.

**Algorithm 2.5:** Simba

1 initialize $\mathbf{w} = (1, 1, \ldots, 1)$;
2 **for** $t = 1$ **to** $T$ **do**
3 pick randomly an instance $\mathbf{x}$ from $S$ ;
4 calculate **nearmis**($\mathbf{x}$) and **nearhit**($\mathbf{x}$) with respect to $S \setminus \{\mathbf{x}\}$ and the weight vector $\mathbf{w}$;
5 **for** $i = 1$ **to** $d$ **do**
6 $\Delta_i = \frac{1}{2} \frac{\partial u(\theta(\mathbf{x}))}{\partial \theta(\mathbf{x})} \left( \frac{(x_i - \mathbf{nearmiss}(x)_i)^2}{\|x - \mathbf{nearmiss}(x)\|_w} - \frac{(x_i - \mathbf{nearhit}(x)_i)^2}{\|x - \mathbf{nearhit}(x)\|_w} \right) w_i$
7 **end**
8 $\mathbf{w} := \mathbf{w} + \Delta$
9 **end**
10 $\mathbf{w} \leftarrow \mathbf{w}^2 / \left\|\mathbf{w}^2\right\|_\infty$ where $(\mathbf{w}^2)_i := (w_i)^2$

The authors present results on a synthetic dataset to illustrate the quality of the margin based evaluation function and the ability of Simba algorithm to deal with dependent features compared to the well known Relief algorithm. The problem consisted of 1000 sample points with 10 real valued features. The target concept is an xor function over the first 3 features. Hence, the first 3 features are relevant while the other features are irrelevant. Notice that this task is a special case of parity function learning and is considered hard for many feature selection algorithms [25]. Thus for example, any algorithm which does not consider functional dependencies between features fails on this task. Figure 2.12 present the results we have been able to reproduce.

They also present results on real data using the AR face database [50] and show that Simba outperforms Relief on this particular classification task and that it handles better correlated features. One of the main advantages of the margin based criterion is the high correlation that it exhibits with the features quality.

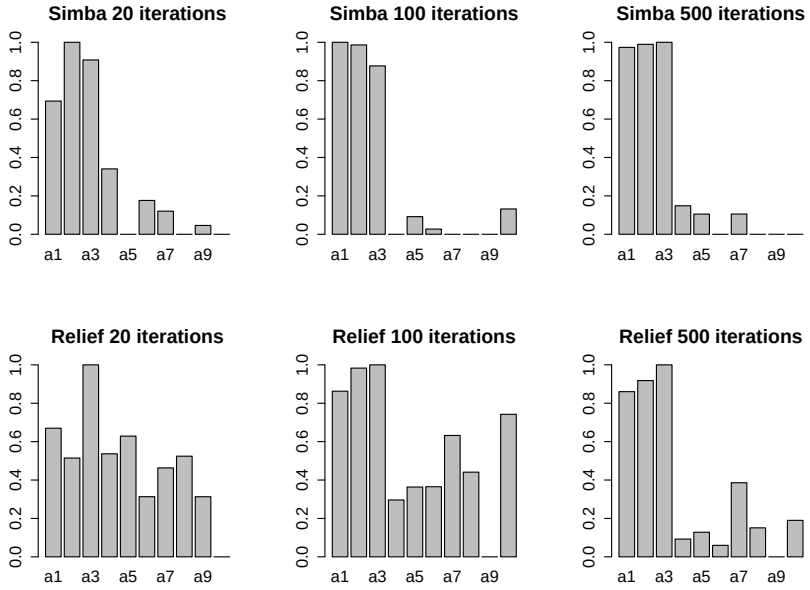



**Figure 2.12:** The weights Simba and Relief assign to the 10features when applying on the xor problem.

Chapter

# 3 Experimental setup

In various sections of this thesis experimental studies are being performed. In order not to repeat the experimental setup in every one of them we present here a description of the used datasets and framework. References are made to this section throughout the document.

## 3.1 Datasets for the experimental studies

Along this thesis various series of experimental work is performed in order to assess the described proposals. Here we describe the datasets that are used along all the experiments. We used datasets from two different sources. First of all we used datasets from the well known UCI repository of machine learning databases [2]. The datasets used can be seen in Table 3.2. Here we present a brief description of each of them.

**Diabetes** Pima Indians Diabetes Data Set. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. There are 2 classes, 768 instances, 8 numeric features.

**Glass** From USA Forensic Science Service this dataset contains 214 instances of different types of glass defined by 10 attributes in terms of their oxide content (i.e. Na, Fe, K, etc). There are 6 classes according to the type of glass (i.e. building windows, vehicle windows, containers,...).

**Heart** Statlog (Heart) Data Set. This database contains 13 attributes from 270 patients. The class attribute indicates the presence of heart disease in the patient.

**Ionosphere** Classification of radar returns from the ionosphere. There are 2 classes, 351 instances, 34 numeric features. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not: their signals pass through the ionosphere.

**Iris** Perhaps the best known database to be found in the pattern recognition literature. The data set contains 4 numeric features and 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

**Landsat** Statlog (Landsat Satellite) Data Set. Multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. 6435 instances with 36 attributes. The aim is to predict this classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number from 1 to 7. There are no examples with class 6 in this dataset.

**LSVT Voice** LSVT Voice Rehabilitation. 126 samples from 14 participants, 309 features. Each attribute corresponds to the application of a speech signal processing algorithm which aims to characterise objectively the signal. Aim: assess whether voice rehabilitation treatment lead to phonations considered 'acceptable' or 'unacceptable' (binary class classification problem).

**Mammogram** Mammography data donated by the Pattern Recognition and Image Modeling Laboratory at University of California, Irvine. There are 86 cases with 65 features each and a binary class indicating benign or malignant.

**Musk** This dataset describes a set of 102 molecules of which 39 are judged by human experts to be musks and the remaining 63 molecules are judged to be non-musks. The 166 features that describe these molecules depend upon the exact shape, or conformation, of the molecule.

**Parkinsons** Oxford Parkinson's Disease Detection Dataset. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

**Pop Failures** Climate Model Simulation Crashes Data Set. Given Latin hypercube samples of 18 climate model input parameter values, predict climate model simulation crashes and determine the parameter value combinations that cause the failures. 540 instances, 18 attributes, 2 classes.

**Spect** The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. There are 22 binary features extracted from the original SPECT images and 267 instances.

**Spectf** The same data as the previous dataset but this time a continuous feature pattern of size 44 was created for each patient. The same binary class and the same 267 instances.

**Sonar** There are 208 patterns obtained by bouncing sonar signals off a metal cylinder and rocks at various angles and under various conditions. Each pattern is a set of 60 numbers

in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The class is binary indicating whether the object was a rock or a metal cylinder.

**Vehicle** Statlog (Vehicle Silhouettes) Data Set. 3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects. The purpose is to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. 18 attributes, 946 instances and 4 classes.

**Waveform** Artificial dataset where each class is generated from a combination of 2 of 3 "base" waves. There are 5000 instances with 21 features each, all of which include noise, and 3 classes.

**Wdbc** Breast cancer databases obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [49]. Features 2 through 10 have been used to represent instances. There are 699 instances with 10 features, each has one of 2 possible classes: benign or malignant.

Another group of datasets that we use along the thesis are the ones used in a feature selection challenge organized by Guyon et al. during the *Neural Information Processing Systems 2003* conference (NIPS 2003) [76]. A summary of the results of this challenge was published by the authors in the next edition of the conference [27]. A number of distractor features called 'probes' having no predictive power had been added to each dataset. A brief description of the 5 datasets is provided below and Table 3.4 show a summary of their characteristics. All details about the preparation of the data are found in the technical report [24]. In all cases we have used both the training and validation sets as we are performing a cross-validated test.

**Arcene** The dataset was obtained by merging three mass-spectrometry datasets to obtain enough training and test data for a benchmark. The samples include patients with cancer (ovarian or prostate cancer), and healthy or control patients. The original features indicate the abundance of proteins in human sera having a given mass value. Based on those features one must separate cancer patients from healthy patients. 200 samples, 10000 variables and 2 classes.

**Dexter** A text classification problem in a bag-of-word representation. Te features represent frequencies of occurrence of word stems in text. The task is to learn which Reuters articles are about 'corporate acquisitions'. 600 samples, 20000 variables and 2 classes.

**Dorothea** This is a drug discovery dataset. Chemical compounds represented by structural molecular features must be classified as active (binding to thrombin) or inactive. 1150 samples, 100000 variables and 2 classes.

**Gisette** This is a handwritten digit recognition problem. The task is to discriminate between to confusable handwritten digits: the four and the nine. The digits have been size-normalized and centered in a fixed-size image of dimension 28x28. The original data were modified for the purpose of the feature selection challenge. In particular, pixels were samples at random in the middle top part of the feature containing the information necessary to disambiguate 4 from 9 and higher order features were created as products of

these pixels to plunge the problem in a higher dimensional feature space. 7000 samples, 5000 variables and 2 classes.

**Madelon** Artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +-1 labels). 2600 samples, 500 variables and 2 classes.

We also use some larger microarray problems. These problems are difficult for several reasons, in particular the sparsity of the data, the high dimensionality of the feature (gene) space, and the fact that many features are irrelevant or redundant. The datasets used can be seen in Table 3.3. We made a preliminary selection of genes on the basis of the ratio of their between-groups to within-groups sum of squares [17]. The best 200 genes for each dataset were selected.

Validation of the described approach uses six public-domain microarray gene expression data sets, shortly described as follows:

1. *Colon Tumor*: Used originally by [1], it consists of 62 samples of colon tissue, of which 40 are tumorous and 22 normal, and contains 2,000 genes.

2. *Leukemia*: Used first by [22], the training set consisted originally of 38 bone marrow examples (plus a further test set with 34 examples). This set of examples has been merged to form a data sample of 72 examples, which are described by 7,129 probes: 6,817 human genes and 312 control genes. The goal is to tell acute myeloid leukemia from acute lymphoblastic leukemia.

3. *Lung Cancer*: Studied by [23], the problem consists in distinguishing between malignant pleural mesothelioma and adenocarcinoma of the lung. There are 181 examples available, described by 12,533 genes.

4. *Prostate Cancer*: This data set was used by [71] to analyze differences in pathological features of prostate cancer and to identify genes that might anticipate its clinical behavior. There are 181 examples and 12,600 genes.

5. *Breast Cancer*: [79] studied 97 patients with primary invasive breast carcinoma; 24,481 genes were analyzed.

6. *GCM*: MIT 14 Global Cancer Map data set, first studied by [62] consists of 190 examples, 16,063 genes and 14 categories corresponding to different malignant tumors.

For comparative purposes, performance results using the whole set of features and the reduced subset of 200 features are displayed in Table 3.1. In view of these results, it is clear that these subsets constitute a very good departing point for further analysis with wrapper methods.

| Problem | 1NN | | LDA | | $SVM_r$ | |
|---|---|---|---|---|---|---|
| | $Y$ | $X_{200}$ | $Y$ | $X_{200}$ | $Y$ | $X_{200}$ |
| Colon Tumor | 23.9 | 23.2 | 24.8 | 20.0 | 31.0 | 14.8 |
| Leukemia | 9.7 | 8.3 | 14.1 | 3.1 | 26.7 | 2.8 |
| Lung Cancer | 1.8 | 2.0 | N/A | 1.8 | 4.4 | 1.0 |
| Prostate Cancer | 23.4 | 19.1 | N/A | 25.5 | 38.2 | 26.9 |
| Breast Cancer | 45.1 | 27.7 | N/A | 24.5 | 48.3 | 24.1 |
| GCM | 10.0 | 13.7 | 13.7 | 10.1 | 5.8 | 5.8 |

**Table 3.1:** Average test error (in %) for the different inducers in the preprocessing phase. $Y$: using the full set of genes; $X_{200}$: using the top pre-selected 200 genes; N/A: computation unaffordable due to numerical inaccuracies in LDA.

## 3.2 Experimental setup for wrappers

We use a carefully designed resampling methodology in order to avoid feature selection bias [63, 72] specially serious in high-dimensional biomedical data, such as gene expression microarrays which are datasets widely used throughout this work. In his paper, Dietterich [15] shows a taxonomy of statistical questions in machine learning. As shown in **Figure.3.1**.

The paper focuses on the boxed node (Question 8). This is also the situation in most of the problems used for the experimental work in this thesis. This is the situation where we are comparing the prediction accuracy of set of algorithms when trained with a data set of a small sample size $S$. Because $S$ is small it will be necessary to use holdout and resampling methods. In the above cited paper, five methods are compared to assess this question: McNemar's test,

**Table 3.2:** UCI dataset descriptions

| problem | features | classes | examples |
|---|---|---|---|
| Diabetes | 8 | 2 | 768 |
| Glass | 10 | 6 | 214 |
| Heart | 13 | 2 | 20 |
| Ionosphere | 34 | 2 | 351 |
| Landsat | 36 | 6 | 6,435 |
| LSVT Voice | 309 | 2 | 126 |
| Mammogram | 65 | 2 | 86 |
| Musk | 168 | 2 | 6,598 |
| Parkinsons | 23 | 2 | 197 |
| Pop Failures | 18 | 2 | 540 |
| SpectF | 44 | 2 | 267 |
| Sonar | 60 | 2 | 208 |
| Vehicle | 18 | 4 | 946 |
| Waveform | 21 | 3 | 5,000 |
| Wdbc | 10 | 2 | 699 |

**Table 3.3:** Microarray dataset descriptions

| problem | features | classes | examples |
|---|---|---|---|
| ma_breast_cancer | 24,481 | 2 | 97 |
| ma_colon_tumor | 2,000 | 2 | 62 |
| ma_gcm | 16,063 | 14 | 190 |
| ma_leukemia | 7,129 | 2 | 72 |
| ma_lung_cancer | 12,533 | 2 | 181 |
| ma_prostate_cancer | 12,600 | 2 | 136 |

**Table 3.4:** NIPS 2003 feature selection challengs dataset descriptions

| problem | features | classes | examples |
|---|---|---|---|
| Arcene | 10,000 | 2 | 200 |
| Dexter | 20,000 | 2 | 600 |
| Dorothea | 100,000 | 2 | 1,150 |
| Gisette | 5,000 | 2 | 7,000 |
| Madelon | 500 | 2 | 2,600 |

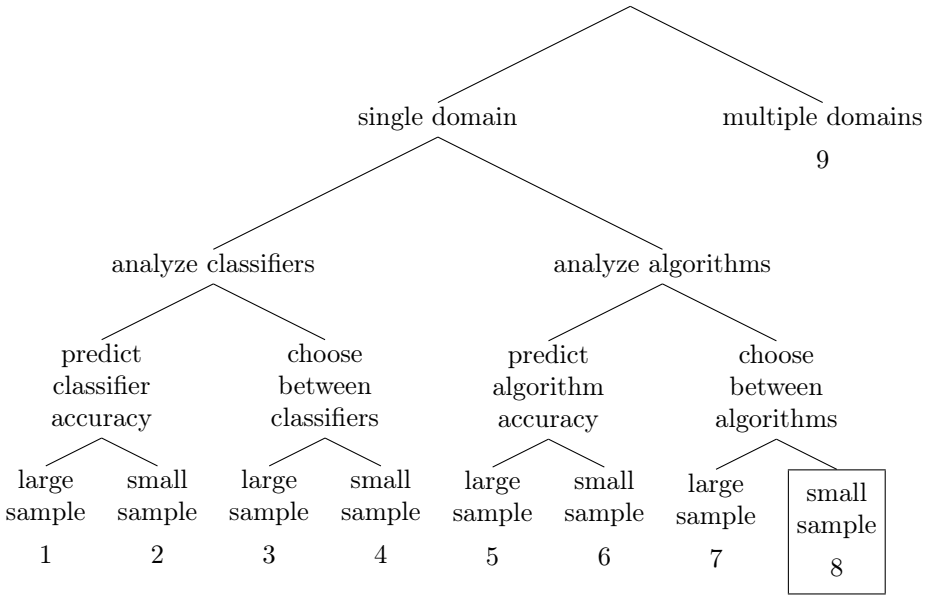



**Figure 3.1:** A taxonomy of statistical questions in machine learning.

a test for the difference of two proportions, the resampled $t$ test, the cross-validated $t$ test and a new test called the 5x2cv test. They show that the best test to minimize Type I error is the proposed 5x2cv test so this will be the one we will be using in all of the experiments where we the conditions for Question 8 are met. It shows that this test performs better than the classic 10-fold cross-validated $t$ test proposed by Kohavi [40]. It was found that the problem of $k$-fold cross-validated $t$ test was too large in some cases. The numerator of the $t$ statistic estimates the mean difference in the performance of the two algorithms over the $k$ folds, while the denominator estimates the variance of these differences. It is stated that the variance was slightly underestimated when the training sets overlapped and the means were occasionally poorly estimated. Moreover, if we replaced the numerator of the $t$ statistic with the observed difference from a *single* fold, the statistic would become well-behaved: this lead to the 5x2cv test.

To compute this test we have to perform 5 replications of 2-fold cross-validation. In each replication, the available data is randomly partitioned into to equal sized sets. Each algorithm is trained using one of the sets and tested using the other one. In our case we want to have different runs of the feature selection process to assess their stability and then test the results with a learner to assess the prediction power of the resulting feature set. So we need two nested loops of this 5x2 cross-validation. This requirement of two nested cross-validation loops further discards 10-fold cross-validation as for problems with a few hundreds of instances such as the gene expression microarray problems we are using the test could be with only one or two instances. **Figure 3.2** shows a graphical representation of the setup.

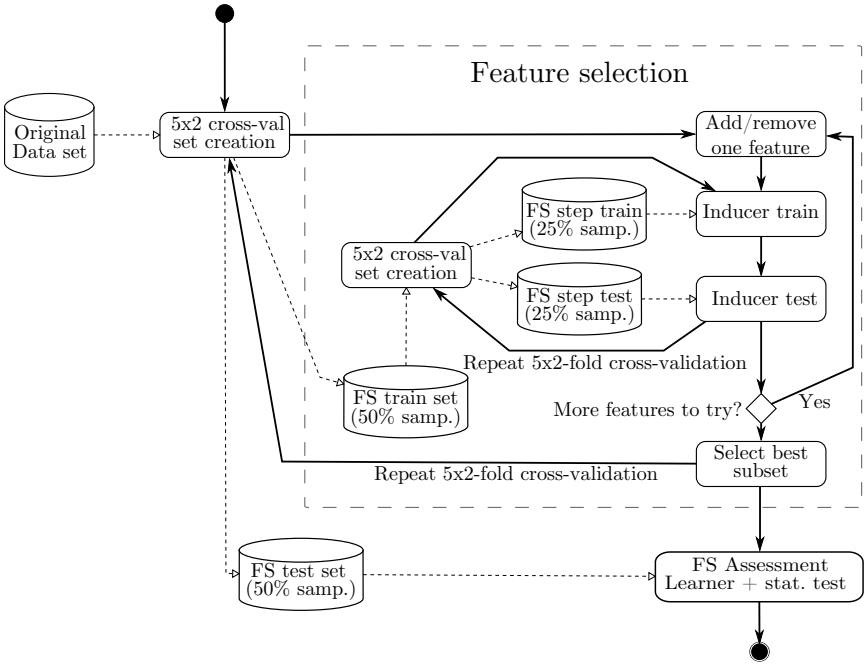



**Figure 3.2:** Graphical representation of the experimental setup

Chapter

# 4 On redundancy and importance

As we mentioned in Chapter 1, one of our hypothesis is that having redundant features in a dataset may negatively affect feature selection algorithms stability. To test this hypothesis we first give a formal definition of feature redundancy. Different definitions can be found in literature. We will describe them along with the problems these definitions present. In fact we propose to use a *level* of redundancy since most of the time we will not find two completely redundant features but a feature that is redundant to a set of other features to a certain degree. In addition, we provide initial work on the definition and study of feature importance. Although the characterizations are of theoretical interest, we do not provide as yet practical algorithms to compute them.

## 4.1 Problems of previous definitions

In general, the definitions of redundancy found in the literature are based on feature correlation, i.e. two features are redundant if their values are correlated. One interesting particular case is when one feature is an exact copy of another so their values are completely correlated, one feature is obviously redundant. But in reality a feature may not be completely correlated with another feature but may be (partially) correlated with a set of features. In such a case it is not straightforward to determine the redundancy. We can take as an example the features shown in Table 4.1. The feature $f_r$ is intuitively redundant with the set $f_1, f_2$ but is not correlated with any of them, so it would not be redundant according to the correlation based definition of redundancy. Therefore, we have to find a better definition for feature redundancy that enables us to identify not only pairs of redundant features but features redundant with any set of other features.

As for feature importance, although there are many metrics in the literature aimed at ranking features –some of them being classifier-independent–, to the best of our knowledge there is no previous attempt to give a formal definition.

| $f_1$ | $f_2$ | $f_r$ | $C$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Table 4.1:** Two relevant features and one redundant: $C = f_1 \wedge f_2$ and $f_r = \overline{f_1 \wedge f_2}$

## 4.2 Redundancy definition

### 4.2.1 Markov blankets

Before giving the formal definition of redundancy let us introduce some previous definitions:

**Definition 4.1.** Let $\mathbf{U} = \{\alpha, \beta, \ldots\}$ be a set of discrete variables in a problem domain. Each variable is associated with a set of possible values. A configuration or a **tuple $\mathbf{u}'$ of $\mathbf{U}' \subseteq \mathbf{U}$** is an assignment of values to every variable in $\mathbf{U}'$.

**Definition 4.2.** A **probabilistic domain model (PDM)** $P$ over $\mathbf{U}$ determines the probability $P(\mathbf{u}')$ of every tuple $\mathbf{u}'$ of $\mathbf{U}'$ for each $\mathbf{U}' \subseteq \mathbf{U}$.

**Definition 4.3.** For three disjoint subsets $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z} \subseteq \mathbf{U}$, $\mathbf{X}$ and $\mathbf{Y}$ are said to be **conditionally independent given $\mathbf{Z}$** under $P$, noted $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_P$ or simply $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ from now on, if (see [55, pp 83–97])

$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \equiv P(\mathbf{x}|\mathbf{y}, \mathbf{z}) = P(\mathbf{x}|\mathbf{z}) \quad \text{whenever } P(\mathbf{y}, \mathbf{z}) > 0 \tag{4.1}$$

Using this notation we can express unconditional independence as $I(\mathbf{X}, \emptyset, \mathbf{Y})$, i.e.,

$$I(\mathbf{X}, \emptyset, \mathbf{Y}) \equiv P(\mathbf{x}|\mathbf{y}) = P(\mathbf{x}) \quad \text{whenever } P(\mathbf{y}) > 0$$

Note that $I(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ implies the conditional independence of all pairs of variables $\alpha \in \mathbf{X}$ and $\beta \in \mathbf{Y}$, but the converse is not necessarily true.

**Definition 4.4.** A **Markov Blanket $\mathbf{BL}_I(\alpha)$** of an element $\alpha \in \mathbf{U}$ is any subset $\mathbf{S} \subset \mathbf{U}$ for which (see [55])

$$I(\alpha, \mathbf{S}, \mathbf{U} - \mathbf{S} - \alpha) \text{ and } \alpha \notin \mathbf{S}. \tag{4.2}$$

An intuitive interpretation of Def. 4.3 would be: Once $\mathbf{Z}$ is given, the probability of $\mathbf{X}$ will not be affected by the discovery of $\mathbf{Y}$. Or $\mathbf{Y}$ is irrelevant to $\mathbf{X}$ once we know $\mathbf{Z}$. Note that the Markov blanket condition in Def. 4.4 is stronger than conditional independence. It is saying that not only that knowing $\alpha$ is irrelevant to the class, but also to the rest of the features, so $\mathbf{S}$ has all the information that $\alpha$ has about $C$ and all the information $\alpha$ has about $\mathbf{U} - \mathbf{S} - \alpha$. This takes us to our definition of redundancy:

**Definition 4.5.** Given a set of features $\mathbf{F}$ and a class feature $C$, a **redundant feature** $\alpha \in \mathbf{F}$ is a feature for which exists a Markov blanket $\mathbf{S} = \mathbf{BL}_I(\alpha)$ within $\{\mathbf{F}, C\}$ such that $\mathbf{S} \subset \mathbf{F}$.

An interesting property of Markov blankets is that if we removed a feature $\alpha$ such that existed $\mathbf{BL}_I(\alpha) \subset \mathbf{U}$ and now we are eliminating another feature $\beta$ such that exists $\mathbf{BL}_I(\beta) \subset \mathbf{U} - \alpha$ then we can prove that also exists $\mathbf{BL}_I(\alpha) \subset \mathbf{U} - \beta$, we can see the proof in [42]. That is, a redundant feature remains redundant when other redundant features are removed. So if we proceed to remove features using this criterion, we will never have to reconsider our decisions.

A PDM $P$ satisfies the following axioms:

- Symmetry:
$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \Leftrightarrow I(\mathbf{Y}, \mathbf{Z}, \mathbf{X})$$
- Decomposition:
$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \Rightarrow I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \wedge I(\mathbf{X}, \mathbf{Z}, \mathbf{W})$$
- Weak union:
$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W}) \Rightarrow I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y})$$
- Contraction:
$$I(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \wedge I(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \Rightarrow I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$$
- Intersection (holds when P is strictly positive, i.e. $P(\mathbf{u}') > 0$, for each tuple $\mathbf{u}'$ of each $\mathbf{U}' \subseteq \mathbf{U}$):
$$I(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \wedge I(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \Rightarrow I(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$$

### 4.2.2 Redundancy level

Unfortunately, we would rarely find a fully redundant feature, but rather one that its information is nearly subsumed by other features. So we would like to know not only whether a feature is redundant or not but its redundancy grade. We would like a function $R'$ which given an feature $\alpha \in \mathbf{U}$ and a set of features $\mathbf{U} \in \mathcal{U}$ gives us a degree of redundancy of this feature to the set. Ideally we would like a function $R' : \mathbf{U} \times \mathcal{U} \rightarrow [0, 1]$ than satisfies the following conditions:

$$R'(\alpha, \mathbf{BL}_I(\alpha)) = 1$$
$$R'(\alpha, \mathbf{U} - \alpha_i) \leq R'(\alpha, \mathbf{U}), \forall \alpha_i \in \mathbf{U}$$

To achieve this we should change the boolean definition of conditional independence to a some function of $P(\mathbf{x}|\mathbf{y}, \mathbf{z})$ and $P(\mathbf{x}|\mathbf{z})$.

**Definition 4.6.** Consider we have that $\mathbf{U}$ is our set of features, $\alpha$ is the feature we are evaluating, and $\mathbf{S}$ is some subset of $\mathbf{U}$ not containing $\alpha$. We defined $\mathbf{u}$ as a configuration of $\mathbf{U}$. We will write $\mathbf{s_u}$, $\mathbf{s_u^{-1}}$ and $\alpha_\mathbf{u}$ for the configuration of $\mathbf{S}$, the configuration of $\mathbf{U} - \mathbf{S} - \alpha$ and the value of $\alpha$ respectively when the configuration of $\mathbf{U}$ is $\mathbf{u}$. Now we can define $\mathcal{U}$ as the

set of all possible configurations of $\mathbf{U}$ for which $P(\mathbf{u} - \mathbf{s_u} - \alpha_\mathbf{u}, \mathbf{s_u}) > 0$. With all that, we define **Redundancy level** $R'$ as:

$$R'(\alpha, \mathbf{U}) = 1 - \max_{\mathbf{S} \subset \mathbf{U} - \alpha} \left( \frac{\sum_{\mathbf{u} \in \mathcal{U}} \left| P(\alpha_\mathbf{u} | \mathbf{s_u}) - P(\alpha_\mathbf{u} | \mathbf{s_u^{-1}}, \mathbf{s_u}) \right|}{|\mathcal{U}|} \right) \tag{4.3}$$

The calculation of this redundancy level is exponential in the number of features in our set, as it compares the conditional probabilities of all possible subsets of $\mathbf{U}$, so the max function will have to compare $|\mathcal{P}(\mathbf{U})| = 2^{|\mathbf{U}|}$ terms. For each subset we also have an exponential cost in the number of values of the features, because the sum is over each configuration $\mathbf{u}$ of $\mathbf{U}$. It is clear to see that, although Eq. 4.3 gives an intuitively consistent definition of redundancy level, its computational cost might be too large for $R'$ to be directly applied in a feature weighting (or feature selection) algorithm. We should then use an estimation of $R'$ that maximized the tradeoff between accuracy and complexity. But in fact the aim of the definition of $R'$ was not to have an efficient algorithm to calculate the redundancy level of a feature. The definition had three basic (related) objectives: first of all to provide a suitable formal definition of redundancy in order to study the effect of feature redundancy in the different existing algorithms, for instance ReliefF. Second, to serve as a starting point for new extensions to methods which performance decreases in the presence of redundant features –again *Relief* is an example. Finally, to direct the development of new algorithms that effectively and efficiently estimate redundancy.

## 4.3 Importance definition

The Bayes error rate is the lowest possible error rate for any classifier of a random outcome and gives a statistical lower bound on the error achievable for a given classification problem and associated choice of features [21].

### 4.3.1 The Bayes Error

In classification, one interested in determining the class or category of objects according to $\Omega$, a discrete random variable taking values in the finite set $\{\omega_1, \ldots, \omega_K\}$ that represent the possible classes, with probabilities $P(\omega_1), \ldots, P(\omega_K)$ acting as *priors*. If the objects are described by real-valued vectors, considering random vectors $X = (X_1, \ldots, X_n)$ with p.d.f. $p(\mathbf{x})$ that measure continuous features of the objects. Let also $\mathcal{P}$ be the support of $p$, i.e. $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^d | \ p(\mathbf{x}) > 0\}$.

In this setting, $p(\mathbf{x}|\omega_i), i = 1, \ldots, K$ are the conditional densities of $\mathbf{x}$ for every class. Then, according to Bayes formula, the *posterior* probabilities are:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{\sum_{i=1}^{m} p(\mathbf{x}|\omega_i)P(\omega_i)}, \text{ with } \sum_{j=1}^{K} P(\omega_j|\mathbf{x}) = 1$$

The classifier that assigns a vector $\mathbf{x}$ to the class with the highest posterior is called the *Bayes classifier*. The error associated with this classifier (more technically, the probability of error) is called the *Bayes error*, which is expressed [21]:

$$P_e = 1 - \sum_{j=1}^{K} P(\omega_j) \int_{R_j} p(\mathbf{x}|\omega_j)\, d\mathbf{x}$$

where $R_j$ is the region in $\mathcal{P}$ where class $j$ has the highest posterior.

### 4.3.2 A definition of feature importance

Denote $X = (X_1, \ldots, X_n)$ the full set of available features. We first consider the Bayes error for a restricted set of features:

**Definition 4.7** (restricted Bayes error)**.** Given $X$, the full set of features, the restricted Bayes error of a subset $X_0 \subseteq X$ is:

$$P_e(X_0) = 1 - \sum_{j=1}^{K} P(\omega_j) \int_{R_j} p_0(\mathbf{x}|\omega_j)\, d\mathbf{x}$$

where $p_0$ is the restriction of $p$ to $X_0$.

**Definition 4.8** (relevant subset)**.** Given $X$, the full set of features, the relevant subset for $X$ is the smallest non-empty $X^*$ such that

$$\forall \hat{X} \subseteq X^*,\ P_e(X^* \setminus \hat{X}) > P_e(X^*)$$

Notice that the relevant subset always exists but is not necessarily unique; in case more than one relevant subset for $X$ might exist, a feasible approach would be to choose the one minimizing a general cost function, as follows: let $c(x) \geq 0$ represent the *cost* of variable $x$ and call $c(\hat{X}) = \sum_{x \in \hat{X}} c(x)$, for any $\hat{X} \subseteq X$. It is assumed here that $c$ is additive, that is, $c(X' \cup X'') = c(X') + c(X'')$ (together with non-negativeness, this implies that $c$ is monotone). The idea is then to look for the relevant subset of smallest cost; when $c(x) = |x|$, then cost is size, and the standard selection setting is recovered. In general, other criteria would be possible, like measurement cost, needed technical skill, etc, depending on the domain at hand. Also noteworthy is the fact that $X^* = X$ is possible, in which case no feature selection is possible without a degradation in accuracy.

**Definition 4.9** (feature importance)**.** Given $X$, the full set of features, the importance $R$ of a feature $X_i \in X$ is:

$$R(X_i) = \begin{cases} P_e(X^* \setminus \{X_i\}) - P_e(X^*) & \text{if } X_i \in X^*, \\ 0 & \text{otherwise.} \end{cases}$$

where $X^*$ is the relevant subset for $X$; note that $R(X_i) \geq 0$, by construction.

**Definition 4.10** (normalized feature importance)**.** Given $X$, the full set of features, the normalized importance $r$ of a feature $X_i \in X$ is:

$$r_i^* = \frac{R(X_i)}{\sum_{j=1}^{d} R(X_j)}$$

Define now, for the sake of simplicity, the importance vector $\mathbf{r}^* = (r_1^*, \ldots, r_n^*)^T$ –thus this vector contains the true relative importances for the full feature set $X = (X_1, \ldots, X_n)$. If we had an unlimited supply of data and computational resources, we could in principle find the importance vector for a feature selection problem. However, in practice we have a data set containing only a finite number of data points, and consequently we can only hope for an estimation of it. Therefore the *dependence* of the importance vector on the available data set is also of theoretical interest.

Consider now $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ a training data set of length $N$, each multivariate instance $\mathbf{x}_n$ with its corresponding class label $t_n$. In practice, the vector of importances will be calculated (estimated) from the data $D$, yielding the *empirical* importance vector $\mathbf{r}^D = (r_1^D, \ldots, r_n^D)^T$.

**Definition 4.11** (bias)**.** The bias vector of an estimation $\mathbf{r}^D$ of $\mathbf{r}^*$ is:

$$\mathcal{B}(\mathbf{r}^D) = \mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*$$

where the $i$-th component is $\text{bias}(r_i^D) = r_i^* - \mathbb{E}_D[r_i^D]$.

**Definition 4.12** (variance)**.** The variance vector of an estimation $\mathbf{r}^D$ of $\mathbf{r}^*$ is the vector $\mathcal{V}(\mathbf{r}^D)$ whose $i$-th component is $\text{var}(r_i^D) = \mathbb{E}_D[(r_i^* - \mathbb{E}_D[r_i^D])^2]$.

In all cases, the expectations are taken with respect to all datasets $D$ of size $N$. We are interested in studying the mean discrepancy (or square error) between the theoretical and the empirical importance vectors.

**Definition 4.13** (MSE)**.** The mean square error or MSE of an estimation $\mathbf{r}^D$ of $\mathbf{r}^*$ is:

$$\text{MSE}(\mathbf{r}^D) = \mathbb{E}_D\left[\|\mathbf{r}^D - \mathbf{r}^*\|^2\right]$$

We can now state our main result:

**Theorem 4.14.**

$$\text{MSE}(\mathbf{r}^D) = \left\|\mathcal{B}(\mathbf{r}^D)\right\|_2^2 + \left\|\mathcal{V}(\mathbf{r}^D)\right\|_1$$

*where $\|\cdot\|_2$ and $\|\cdot\|_1$ stand for the two- and one-norm, respectively.*

*Proof.* The MSE can be expressed as

$$\begin{aligned}
&\mathbb{E}_D\left[\|\mathbf{r}^D - \mathbf{r}^*\|^2\right] = \mathbb{E}_D\left[\|\mathbf{r}^D - \mathbb{E}_D[\mathbf{r}^D] + \mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*\|^2\right] \\
&= \mathbb{E}_D\left[\|\mathbf{r}^D - \mathbb{E}_D[\mathbf{r}^D]\|^2\right] && (\mathbf{A}) \\
&+ \mathbb{E}_D\left[\|\mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*\|^2\right] && (\mathbf{B}) \\
&- 2\,\mathbb{E}_D\big\{(\mathbf{r}^D - \mathbb{E}_D[\mathbf{r}^D])^T(\mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*)\big\} && (\mathbf{C})
\end{aligned}$$

(**B**)

$$\begin{aligned}
&\mathbb{E}_D\left[\|\mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*\|^2\right] \\
&= \mathbb{E}_D\left[\|\mathbf{r}^*\|^2\right] + \mathbb{E}_D\left(\mathbb{E}_D\left[\|\mathbf{r}^D\|^2\right]\right) - 2\,\mathbb{E}_D\left(\mathbb{E}_D[\mathbf{r}^D]^T\mathbf{r}^*\right) \\
&= \|\mathbf{r}^*\|^2 + \mathbb{E}_D\left[\|\mathbf{r}^D\|^2\right] - 2\,(\mathbf{r}^*)^T\mathbb{E}_D[\mathbf{r}^D] \\
&= \|\mathbf{r}^* - \mathbb{E}_D[\mathbf{r}^D]\|^2 = \sum_{j=1}^{d}\left(r_i^* - \mathbb{E}_D[\mathbf{r}^D]_i\right)^2 \\
&= \sum_{j=1}^{d}\left(r_i^* - \mathbb{E}_D[r_i^D]\right)^2 = \sum_{j=1}^{d}(\mathrm{bias}(r_i^D))^2 = \left\|\mathcal{B}(\mathbf{r}^D)\right\|_2^2
\end{aligned}$$

(**A**)

$$\begin{aligned}
&\mathbb{E}_D\left[\|\mathbf{r}^D - \mathbb{E}_D[\mathbf{r}^D]\|^2\right] = \mathbb{E}_D\left[\|\mathbf{r}^D\|^2\right] - \|\mathbb{E}_D\left[\mathbf{r}^D\right]\|^2 \\
&= \sum_{j=1}^{d}\mathbb{E}_D[r_i^D] - \sum_{j=1}^{d}\left(\mathbb{E}_D[\mathbf{r}^D]_i\right)^2 = \sum_{j=1}^{d}\mathbb{E}_D[r_i^D] - \sum_{j=1}^{d}\left(\mathbb{E}_D[r_i^D]\right)^2 \\
&= \sum_{j=1}^{d}\mathbb{E}_D\left[\left(r_i^D - \mathbb{E}_D[r_i^D]\right)^2\right] = \sum_{j=1}^{d}\mathrm{var}(r_i^D) = \sum_{j=1}^{d}|\mathrm{var}(r_i^D)| = \left\|\mathcal{V}(\mathbf{r}^D)\right\|_1
\end{aligned}$$

(**C**)

$$\begin{aligned}
&\mathbb{E}_D\big\{(\mathbf{r}^D - \mathbb{E}_D[\mathbf{r}^D])^T(\mathbb{E}_D[\mathbf{r}^D] - \mathbf{r}^*)\big\} \\
&= \mathbb{E}_D\left[(\mathbf{r}^D)^T\mathbb{E}_D[\mathbf{r}^D]\right] - (\mathbf{r}^*)^T\mathbb{E}_D[\mathbf{r}^D] - \|\mathbb{E}_D[\mathbf{r}^D]\|^2 + (\mathbf{r}^*)^T\mathbb{E}_D[\mathbf{r}^D] \\
&= \mathbb{E}_D[\mathbf{r}^D]^T\mathbb{E}_D[\mathbf{r}^D] - \|\mathbb{E}_D[\mathbf{r}^D]\|^2 = \|\mathbb{E}_D[\mathbf{r}^D]\|^2 - \|\mathbb{E}_D[\mathbf{r}^D]\|^2 = 0
\end{aligned}$$

■

We see that the expected squared difference between the theoretical and the empirical importance vectors can be expressed as the sum of two terms. The first term represents the extent to which the average estimation over all data sets differs from the theoretical one. The second term measures the extent to which the point estimations for specific data sets vary around their average, thus measuring the stability of the estimation in relation to a particular choice of data.

On the other hand, the definition and study brings two distinctive advantages: first, it is not dependent on the number of dimensions $d$ or classes; second, although the focus is in classification problems, the adaptation to regression problems would be straightforward,

simply replacing the Bayes probability of error by the corresponding optimal error measure (e.g., the error committed by the *regression function* in the regression case). The presented bias/variance analysis turns out to be independent of the problem being a classification or a regression one.

Chapter

# 5 A focus on Relief

In this chapter we present an on-depth study on a the popular Relief algorithm. We review characteristics such as the metric it uses or its robustness against feature redundancy. We also propose some modifications to improve it.

## 5.1 Study of Relief metric

Relief needs a heterogeneous metric to be able to handle both continuous and nominal attributes. As we have seen in Chapter 2, the metric used in the original algorithm was the Heterogeneous Euclidean-Overlap Metric (HEOM) which uses the euclidean distance (2.20) for continuous attributes and *overlap* (2.19) for the nominal ones.

As Wilson and Martinez [83] pointed out, this approach does not take into account all the information nominal attributes are providing. They are skipping some information that may later be useful in the learning process.

To overcome this problem Stanfill and Waltz [74] introduced the Value Difference Metric (VDM) to provide an appropriate distance function for nominal attributes. VDM defines the distance between two instances as:

$$D_{VDM}(\mathbf{x}, \mathbf{y}) = \sum_{a \in A} d_{VDM}\left(x_a, y_a, a\right) w\left(a, x_a\right) \tag{5.1}$$

where,

$$d_{VDM}(x, y, a) = \sum_{c \in C} \left( \frac{N_{a=x,c}}{N_{a=x}} - \frac{N_{a=y,c}}{N_{a=y}} \right)^2 \tag{5.2}$$

$$w(a, x) = \sqrt{\sum_{c \in C} \left( \frac{N_{a=x,c}}{N_{a=x}} \right)^2} \tag{5.3}$$

$N_{a=x,c}$ is the number of instances of class $c$ which have the $x$ as the value for attribute $a$ and $N_{a=x}$ is the total number of instances which have $x$ as the value for attribute $a$.

We can also express the VDM in a simpler way by using conditional probabilities:

$$d_{VDM}(x,y,a) = \sum_{c \in C} \left[P\left(c|x_a\right) - P\left(c|y_a\right)\right]^2 \tag{5.4}$$

$$w(a,x) = \sqrt{\sum_{c \in C} \left[P\left(c|x_a\right)\right]^2} \tag{5.5}$$

Where $P\left(c|x_a\right)$ is de conditional probability of one instance to be of class $c$ knowing that its attribute $a$ has a value of $x$.

The factor $w(a,x)$ is the weight of an attribute and tries to bring information about the discrimination power of this attribute. The minimum value of this weight represents a uniform distribution of the attribute values among the different classes.

$$w(a,x) = \sqrt{\sum_{c \in C} \frac{1}{|C|^2}} = \sqrt{\frac{|C|}{|C|^2}} = \sqrt{\frac{1}{|C|}} = |C|^{-1/2} \tag{5.6}$$

And will reach its maximum value when $a$ is a perfect discriminator – when value $x$ only appears in instances belonging to one class. In addition we can easily see that this maximum value is 1. This metric is not exempt of problems. As we have seen this metric only takes into account the conditional probability distributions of the attribute values given a the class. This will make two attributes with the same conditional probability distribution with respect to the class will be at 0 distance according to this metric. This might not be interesting in some cases. This is specially true in problems with two attributes with attributes having near uniform distribution given the class when taken individually but that are good discriminators when combined. The *parity-n* problem is a good example. This problem consists of $n$ binary attributes. The class will be 1 for instances with an even number of attributes with a value of 1 and 0 otherwise. Every attribute has a uniform distribution given the class but the combination of all of them constitute a perfect discriminator of the class. In this case the VDM distance between two random instances will always be 0 provided that all the conditional probabilities will have the same value so giving no information about which is the nearest neighbour for a given instance. Moreover, if we add irrelevant attributes also uniformly distributed given the class, they will also have the same conditional probability distribution. Therefore the difference among two instances using this attributes will also be 0 and so RELIEF will assign them exactly the same weight assigned to the relevant attributes. We can see that for problems with attributes that are high interactions using the VDM is probably a bad idea.

## 5.2 Redundancy analysis

RELIEF gives us a measure of the attributes relevancy. Our aim is to study what happens when attributes with a high level of redundancy by answering the following questions:

1. Will it give the same weight to two redundant variables?

2. Can we conclude that two variables are redundant if they have the same weight?
3. Do redundant variables harm each other?

Let's assume we have two attributes $A_1$ and $A_2$ which are completely redundant to each other, they allow a learner to predict the class for the same instances. To answer the first question we will assume we have a problem with these two attributes and a third one (which will be called $A_3$)) which is not redundant to the other two. We only have to prove that the difference between the *nearest hit* and the *nearest miss* is the same for $A_1$ and $A_2$. If this was true the weight update at each iteration of RELIEF would also be equal and at the end $A_1$ and $A_2$ would be assigned the same weights.

**Lemma 5.1.** *If two variables are completely redundant to each other* RELIEF *will assign the same weights to them.*

*Proof.* Let's start with a demonstration for two-classes problems, binary attributes and no noise and then we will generalize it. In the given situation we have the following values:

**Case 1:** $A_1 = X_1\ A_2 = X_1\ A_3 = Y_1\ C = Z_1$

**Case 2:** $A_1 = X_1\ A_2 = \neg X_1\ A_3 = Y_1\ C = Z_1$

A otherwise the attributes would not be useful to determine the class in for the same instances. The *nearest hit* and the *nearest miss* would be:

**Case 1:** $A_1 = X_1'\ A_2 = \neg X_1'\ A_3 = Y_1'\ C = Z_1'$

Where $Z_1'$ will be $Z_1$ for the *nearest hit* and $\neg Z_1$ for the *nearest miss*. If $X_1'$ have a different value than $X_1$ then the difference between the two attributes will be 1 and if they have equal values will be 0.

**Case 2:** $A_1 = X_1'\ A_2 = \neg X_1'\ A_3 = Y_1'\ C = Z_1'$

If $X_1'$ is different from $X_1$ then $\neg X_1'$ will also be different from $\neg X_1$ and the difference will be 1, otherwise both will be 0. The extension to non-binary attributes is fairly simple: the attributes will have more possible values, but if $A_1$ has the same values for the *nearest hit* and the *nearest miss*, $A_2$ will unavoidably also have the same value to meet the redundancy hypothesis. And the same applies to different values. For problems with more that two classes all the above also holds true: for each *nearest misses* the relationship between $A_1$ and $A_2$ is still the same so the weight increment will be the same ■

Now we will consider the case of two attributes that have a level of redundancy according to Eq. 4.3 but are not completely redundant by introducing some noise in its values. Let's assume that the probability of a certain value to be affected by noise is $P_n$. Let's see how this noise affects the difference of a certain instance $\mathbf{x_1}$ with its *nearest hit* and its *nearest miss* for an attribute $A$ with a replica of $A$ that we will call $A'$ and that is affected by noise with a

probability $P_n$. The following formula computes the difference increase from $\mathbf{x_1}$ to any other instance $\mathbf{x_2}$.

$$pdc_{A-A'} = p(V_1 = V_2) \wedge p(V_1' \neq V_2') + p(V_1 \neq V_2) \wedge p(V_1' = V_2') \tag{5.7}$$

Where $pdc_{A-A'}$ is the probability of a difference change between instances $\mathbf{x_1}$ and $\mathbf{x_2}$ for attributes $A$ and $A'$, $V_1$ and $V_2$ are the values for $\mathbf{x_1}$ and $\mathbf{x_2}$ respectively for $A$ and $V_1'$ and $V_2'$ the values for $A'$. As $A$ and $A'$ are not independent, we can rewrite the formula as:

$$pdc_{A-A'} = p(V_1' = V_2'|V_1 \neq V_2)p(V_1 \neq V_2) + p(V_1' \neq V_2'|V_1 = V_2)p(V_1 = V_2) \tag{5.8}$$

By further developing the above formula, we can compute the probability of $V_1'$ to be equal to $V_2'$ knowing that $V_1$ and $V_2$ were different. We will consider various cases: if $V_1'$ has changed but $V_2'$ no, we know that the probability of them being equal knowing that they differed before is of one in the number of possible values minus one. We have an analogous case if the one that changed is $V_2'$ but $V_1'$ stayed the same. If both change then the probability of them having the same value is the probability of the one change not to pick the same value of the other one as we are assuming that both are changing. That makes the probability to be again of one in the number of possible values except the previous one $|V| - 1$. We defined the probability of changing a certain value to be $P_n$ so we already have the first part of the sum broken down. Now we analyze the reverse case: the probability of $V_1'$ to differ from $V_2'$ knowing that $V_1$ and $V_2$ were equal. This is a simpler situation. If only one of them varies the difference will also vary. If both of them vary then the probabilty of the difference to vary is the probability of the two variables not to pick the same value. That is: $|V| - 2$ divided by $|V| - 1$, as once we pick a value for one of them that is the proportion of cases when the second one will have a different value. Now we can rewrite the formula:

$$pdc_{A-A'} = \frac{2\left(ps\left(1-ps\right)\right)}{|V|-1} + \frac{ps^2\left(|V|-2\right)}{\left(|V|-1\right)^2} p(V_1 \neq V_2) + 2\left(ps\left(1-ps\right)\right) + \frac{ps^2\left(|V|-2\right)}{\left(|V|-1\right)} p(V_1 = V_2) \tag{5.9}$$

A specific case for this formula is when $|V|$ is 2 (i.e. a binary attribute). In this case, the two terms that multiply the probability of $V_1$ and $V_2$ to be different and to be equal will be the same, so we can simplify the formula and knowing that the sum of the two is 1 (they either have equal values or different values), we end up with a formula as simple as:

$$pdc_{A-A'} = 2\left(ps\left(1-ps\right)\right) = \left(2ps - 2ps^2\right) \tag{5.10}$$

Let's now study the effect to the weight increment. To simplify, let's call $\mathcal{D}$ to the factor multiplying the probability of $V_1$ and $V_2$ to be different and $\mathcal{E}$ to the factor multiplying the probability of them being equal. We have:

$$pdw_{A-A'} = \mathcal{D} \cdot p(V_1 \neq V_2)_{NM} \mathcal{E} \cdot p(V_1 = V_2)_{NM} - (\mathcal{D} \cdot p(V_1 \neq V_2)_{NH} \mathcal{E} \cdot p(V_1 = V_2)_{NH}) \tag{5.11}$$

We just applied the above ideas to the formula that RELIEF uses for the weight increments: the *nearest miss* difference adds to the weight (we seek attributes that make instances of other

classes to be far) while the *nearest hit* subtract form the weight (we seek attributes that make instances of the same class near). Simplifying:

$$pdw_{A-A'} = X(p(V_1 \neq V_2)_{NM} - p(V_1 \neq V_2)_{NH}) + Y(p(V_1 \neq V_2)_{NH} p(V_1 \neq V_2)_{NM}) \quad (5.12)$$

A first observation is that this formula, in the presented case where the attributes are binary (i.e. $|V| = 2$), as $\mathcal{D} = \mathcal{E}$ and the other terms cancel each other, we can deduct that the added noise will not affect the weight calculation. For other values of $|V|$ Fig. 5.1 shows the importance of each term. We can see that as the relationship between the probabilities

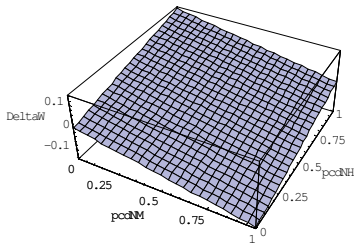



**Figure 5.1:** $P_n = 0.1$, $|V| = 3$

of the values to be different for the *nearest hit* and*nearest miss* increase, the weight change probability also increases. This makes the noise to have more effect for variables which have high values for this relationship. This means that noise will decrement the weights for attributes where the values for instances and their *nearest hits* are often equal while difference in values for instances and their *nearest misses* are often different (i.e. the most relevant attributes). For attributes in the opposite situation (i.e. not that relevant as they had negative contribution form the *nearest hits*) noise will increment their weights. The conclusion is that noise will make the weights go to zero for both relevant and irrelevant features Let's see which is the contribution of the probability of noise $P_n$. In Fig. 5.2 we can see that $P_n$ is

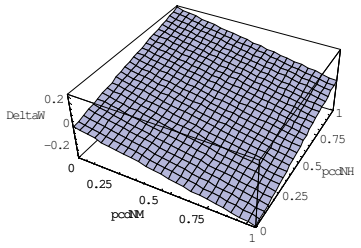



**Figure 5.2:** $P_n = 0.2$, $|V| = 3$

only controlling the range of the probability of the weight increment to vary.

Now we will study the inverse implication that we stated in question 2 above: Can we conclude that two variables are redundant if they have the same weight?

**Lemma 5.2.** *Two attributes that are assigned the same weight by* RELIEF *are not necessarily redundant.*

*Proof.* It seems fairly obvious that the answer will be no. As we stated above RELIEF weights are correlated with attributes contribution to te correct discrimination of the class by a learner. Based on this, two attributes that are useful to determine the class of an instance the same number of times but for completely different instances will be given the same weight. We can give a simple counter-example: The extension to multiple variables of the XOR problem (*parity-n*). In this problem the *nearest hits* will always subtract (instances of the same class may have a different value in every attribute), and *nearest misses* will add or subtract depending on how we break ties. If we break ties randomly it's easy to see that the contribution for *nearest misses* will also be the same for two redundant attributes. This will lead the two attributes to be assigned the same weight even though they are in no way redundant to each other. ■

We will finally answer question 3 where we considered the effect of redundancy to the algorithm performance. We start by studying the changes in the weights RELIEF assigns to attributes before and after adding redundant attributes and then explaining the reasons for that change. We will use the same problem we used before when answering question 1 but first of all only with $A_2$ and $A_3$ and then we will add $A_1$. What we empirically observe is that the weight of $A_2$ has diminished (and we know from question 1 that the weight for $A_1$ will be the same as for $A_2$). The reason why the weight diminishes is that by adding a redundant attribute $A_1$ the *nearest miss* may change and if this change affects the variation in the weight difference then it does it in a negative way: Let's imagine that we have an instance $\mathbf{x_1}$ and that its *nearest miss* $NH(\mathbf{x_1})$ is at distance $d$. If they have the same value for $A_2$, then it's impossible that the *nearest miss* will change by adding $A_1$ as the distance between the two will be the same and if it was the nearest instance of different class it has to keep being it. On the other hand if the two instances had different values for attribute $A_2$, adding $A_1$ may make another instance $\mathbf{x_2}$ to be at the same distance $(d+1)$ as $NH(\mathbf{x_1})$ in that case the new *nearest miss* will have the same value for attribute $A_2$ so it will have a negative impact in the weight calculation. In the same way the contrary holds true for the *nearest hit*: For the same instance $\mathbf{x_1}$ and its *nearest hit*, if both have the same value for $A_2$ then it is impossible that by adding $A_2$ the nearest neighbour chages (using the same reasoning as above). But if they have different values for $A_2$, adding $A_1$ may make another instance to become the *nearest hit* and the new instance will have the same value for $A_2$ so it will have a positive influence in the weight.

With the above statements one could think that the two influences cancel each other but, by definition of how RELIEF works, the more relevant an attribute the lower the differences with near instances if the same class and the higher the differences with near instances of different classes. That causes what we could call a *ceiling effect* to the influence of *nearest hits*: even though the instances of the same class get closer when replicating the attribute the weight variation is very low as it was not penalizing before but when instances of different classes get closer they have a negative effect on the weight. And the same happens with *nearest misses*. The result is that when adding redundant attributes to one of the existing its weight will get closer to 0 as Šikonja and Kononenko [70] pointed out.

## 5.3 Double Relief

When more and more irrelevant features are added to a dataset the distance calculation of Relief degrades its performance as instances may be considered neighbors when in fact they are far from each other if we compute its distance only with the relevant features. In such cases the algorithm may lose its context of locality and in the end it may fail to recognize relevant features.

The $\text{diff}(A_i, \mathbf{x_1}, \mathbf{x_2})$ function calculates the difference between the values of the feature $A_i$ for two instances $\mathbf{x_1}$ and $\mathbf{x_2}$. Sum of differences over all features is used to determine the distance between two instances in the nearest hit and miss calculation (see Eq. 2.18).

As seen in the k-nearest neighbors classification algorithm (kNN) many weighting schemes which assign different weights to the features in the calculation of the distance between instances (see Eq. 5.13).

$$\delta'(\mathbf{x_1}, \mathbf{x_2}) = \sum_{i=1}^{a} w(A_i)\,\text{diff}(A_i, \mathbf{x_1}, \mathbf{x_2}) \tag{5.13}$$

In the same way that in [82] Relief's estimates of features' quality have been used successfully as weights for the distance calculation of kNN we could use their estimation in the previous iteration to compute the distance between instances while searching the nearest hits and misses. We will refer to this version of ReliefF as double ReliefF or in short dReliefF.

### 5.3.1 Progressively weighted double Relief

The problem using the weights estimates could be that in early iterations these estimations could be too biased to the first instances and could be far from the optimal weights. Therefore, for small $t$, $W[A_i]$ is very different from $W[A_i]_t$.

What we want is to begin the distance calculation without using the weight estimates and then, as Relief's weight estimates become more accurate (because more instances have been taken into account), increase the importance of these weights in the distance calculation. Lets have a distance calculation like the one in Eq. 5.14.

$$\delta(\mathbf{x_1}, \mathbf{x_2}) = \sum_{i=1}^{a} f(W(A_i)_t, t)\,\text{diff}(A_i, \mathbf{x_1}, \mathbf{x_2}) \tag{5.14}$$

We would like a function $f : \mathbb{R} \times (0, \infty) \to \mathbb{R}$ such that:

- $f(w, t)$ is increasing with respect to $t$
- $f$ is continuous
- $f(w, 0) = 1$
- $\lim_{t\to\infty} f(w, t) = w$

One such function could be the one in Eq. 5.15. And we will refer to the version of ReliefF using this distance equation as progressively weighted double Relief or in short pdReliefF.

$$f(w,t) = \frac{(w-1)c(t)}{c(t)+s} + 1 \tag{5.15}$$

Where $s$ is a control parameter that determines the steepness and final value of the curve described by $f$ (see Fig. 5.3) and $c(t)$ is a function of the iteration number (e.g. $c(t) = t$). Another desirable property for our function would be that it always gives the same results regardless of the number of iterations. In other words, if $m$ is the total number of iterations, we would like $f(w,m)$ to be the same value whatever the value of $m$. To achieve that we must make $c(t)$ depend also on the total number of iterations $m$ so as to decrement the steepness of the function as the number of total iterations increases. A posible definition of $c(t)$ is shown in Eq. 5.16.

$$c\,(t) = (t/m)^a \tag{5.16}$$

In Fig. 5.4 we can see how $f$ varies the influence of different weights (even a non-realistic one that is greater than 1) as iterations go on. We can see that with high values of $s$ the function converges in the first few iterations and then it stabilizes its value near $w$ and for low values of $s$ it's value remains near 1 till the end. To choose a value we can compute the area left over and below the function. We can see the normal ReliefF as a particular case where $f(w,t) = 1$ having maximum area and dReliefF as another particular case with $f(w,t) = w$ having minimum area. We want to choose the parameters to be in between the two. Specifically we could choose the parameters so as to leave $1/3$ of the area below the function. For doing this we have to solve Eq. 5.17

$$\frac{\int_1^m f(w,t)\,dt - \int_1^m w\,dt}{\int_1^m 1\,dt - \int_1^m w\,dt} = \frac{1}{3} \tag{5.17}$$

A combination of parameters that solves the equation are: $a = 2$ and $s = 0.0633657 \simeq 0.06$. Graphically it can be seen in Fig. 5.4 that those values make weights' ponderations stay near 1 for half of the iterations and then takes values near the weights' values. This value has been chosen in our experiments.

### 5.3.2 Experimental design

#### Objective

The above sections present three algorithms:

**ReliefF** The algorithm presented by Kononenko in [43]

**dReliefF** The above algorithm using it's own partial weigts to ponderate attributes in distance calculation

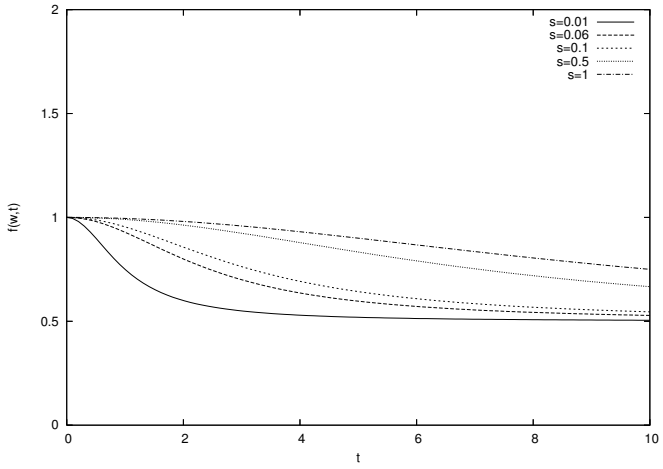



**Figure 5.3:** Plot of function $f$ for 10 instances with $w = 0.5$ and $c(t) = (t/m)^2$

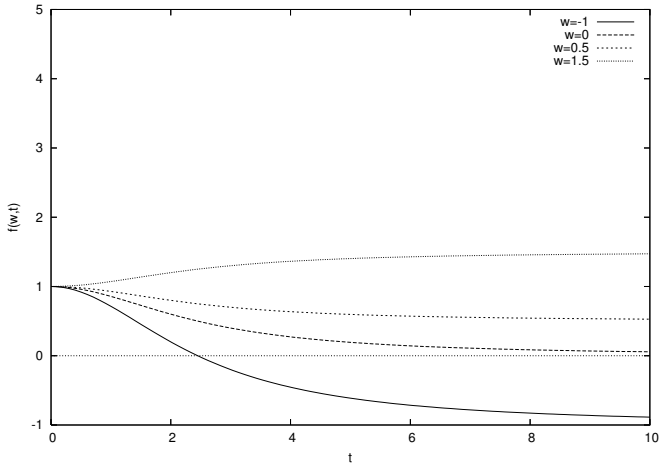



**Figure 5.4:** Plot of function $f$ for 10 instances with $s = 0.06$ and $c(t) = (t/m)^2$

**pdReliefF** The above using a function to progressively increment the weights ponderation effect in distance calculation

The objective of the experiments which will be presented is to compare performance of the three algorithms related to the factor of irrelevant attributes. The hypothesis is that the performance of the non-modified algorithm will be more affected by the number of irrelevant attributes increase due to their influence in distance calculation.

### Factors

As stated before the key factor of the experiments is the ratio of irrelevant attributes, but there are some nuisance factors which have effect on the experiments' results. The factors considered in the experiments are:

- Problem to solve
- Numeric vs. categoric attributes
- Number of relevant attributes
- Number of irrelevant attributes
- Data randomization

The main factor that will impact on performance results will be the problem we want to solve and in addition will be the most difficult to reduce. In order to eliminate it's influence, all the possible problems would have to be tried which is obviously impossible. Another factor that can clearly impact on performance is the type of the attributes as Relief has an heterogeneous function for distance calculation which depends on whether the attributes are numeric or categoric. Hence, to reduce the effect of these two factors the same experiments will be run on six different problems, three with numeric attributes and three with categoric ones. All the problems tested will be artificial to have sufficient knowledge about the data not to make performance of the weighting dependent on performance of a classifier.

Ranges for each factor have to be chosen. There has to be at least one relevant attribute and one irrelevant one in order to check whether the algorithm seems capable of distinguishing them, so both of them will start at 1 in our experiments. The number of irrelevant attributes will depend on the number of relevant ones in order to test with the same percentage of irrelevant attributes for each number of relevant attributes. A good choice could be to have at most twice the number of irrelevant attributes as the number relevant ones.

The upper bound for the number of relevant attributes will depend on the number of instances that are to be generated. It is interesting to test the algorithms with a wide range of attributes to instances ratios. We may arbitrarily set number of instances generated to 100. With that number of instances, it would be interesting to have at most 150 features for the ratio of attributes to instances not to get too low. If we want total features to keep below 150 with a number of irrelevant attributes of twice the number of relevant ones, we have to set upper bound to the number of relevant attributes to 50.

Finally 10 different sets of data will be generated for each combination of other factors to reduce the possible effect of randomly generating a pathologic set of data.

### Design

Here we have to decide which of all the possible combinations of factors will be tried in the experiments. The better way to reduce or eliminate the contribution to experimental error of each of the factors would be to treat them as blocking factors. That is to create homogeneous blocks in which the factors are kept constant while the target factor takes all its possible values. When blocking is not possible because of limited resources a random subset of each block can be run.

With the ranges described above, there are a total of $3 \times I \times N \times (N-1)$ different factor combinations for each problem as seen on Eq. 5.18, where $N$ is the number of relevant attributes and $I$ the number of iterations (i.e. random dataset generations) for each combination of relevant and irrelevant attribute numbers.

$$\left( \sum_{imp=1}^{N} 2imp \right) \times I_{\text{iterations}} \times 3_{\text{algorithms}} = 3 \times I \times N \times (N-1) \tag{5.18}$$

That gives a total number of 76,500 different combinations for each problem. With that number of combinations all combinations can be run. Hence the experimental design will be a full blocking design as shown on Fig. 5.5 in an algorithmic way.

```
1. for each problem in problems do begin
2.     for impAtts := 1 to 50 do begin
3.         for irrAtts := 1 to impAtts * 2 do begin
4.             for iteration := 1 to 10 do begin
5.                 execute problem with each algorithm;
6. end;
```

**Figure 5.5:** Pseudo code of the experimental design

### Problems

#### RDG1NamedContinuous

A data generator that produces data randomly with numeric attributes by producing a decision list. The decision list consists of rules. The rules have the form $c_x := \bigwedge_1^n t$, where $t$ is an inequality term (i.e. $x < y$ or $x \geq y$) between some attribute and a random value. For each rule, the number $n$ will be a random number in the range [1..10]. An example set of rules can be seen on Eq. 5.19.

$$\begin{aligned} &\text{RULE 0: } c_0 := a_1 < 0.986 \wedge a_0 >= 0.65 \\ &\text{RULE 1: } c_1 := a_1 < 0.95 \wedge a_2 < 0.129 \\ &\text{RULE 2: } c_2 := a_1 >= 0.562 \end{aligned} \tag{5.19}$$

Instances are generated randomly one by one. The class will be determined by the first rule that is true for the current instance. If decision list fails to classify the current instance, a new rule according to this current instance is generated and added to the decision list. Irrelevant attributes are generated randomly in the range $[0, 1]$.

**RandomRBFRandRed1**

Radial basis functions (RBF) are functions which characteristic feature is that their response decreases (or increases) monotonically with distance from a central point. There are different formulas to describe the specific shape of the function and they usually have parameters to control the center and the distance scale. In this particular case, the function $f(x)$ used is the Gaussian which is described by Eq. 5.20 and can be seen on Fig. 5.6. Its parameters are its mean $\mu$ and its standard deviation $\sigma$. A Gaussian RBF monotonically decreases with distance from the center.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{5.20}$$

RandomRBF data is generated by first creating a random set of centers for each class. Each center is randomly assigned a weight, a central point per attribute, and a standard deviation. To generate new instances, a center is chosen at random taking the weights of each center into consideration. Attribute values are randomly generated and offset from the center, where the overall vector has been scaled so that its length equals a value sampled randomly from the Gaussian distribution of the center. The particular center chosen determines the class of the instance. RandomRBF data contains only numeric attributes as it is non-trivial to include nominal values. Irrelevant attributes are generated following the same Gaussian distribution for some random centers and standard deviation.

**NonMonotonic**

Let $r_a$ be a random value in the range [0..1] to act as a ponderator for the attribute $a$. Now, for each instance $i$ generate a random value $r_i$ in the rage $[0..N]$, where $N$ is the number of important attributes. The value $a_i$ of the attribute $a$ for instance $i$ will be the one in Eq. 5.21.

$$a_i = \begin{cases} r_a \times r_i & \text{if } (i \mod 2) \neq 0 \\ r_a \times \sqrt{r_i} & \text{if } (i \mod 2) = 0 \end{cases} \tag{5.21}$$

The class for instance $i$ will be the integer part of $r_i$. Irrelevant attributes are created randomly following a uniform distribution in the range $[0, 1]$.

**MajorityN**

Creates $n$ binary attributes and $i$ irrelevant attributes. The class attribute is 1 when the instance has a majority of 1s in the relevant attributes and 0 otherwise.

**ModuloP**

Each Modulo-p problem is described by a set $|\mathcal{R}| = n$ of relevant attributes and $i$ irrelevant attributes, both with integer values in the range $[0, p)$. The class $c$ can be defined as in Eq.

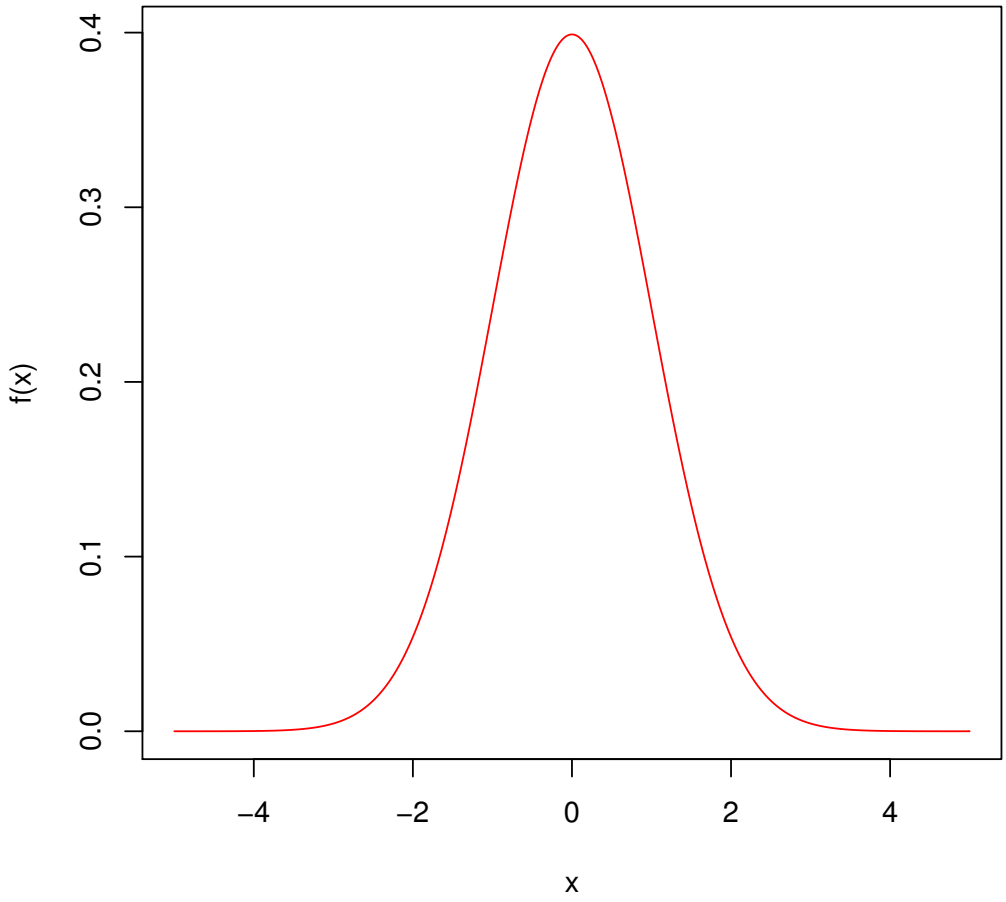



**Figure 5.6:** Plot of function $f(x)$ with $\mu = 0$ and $\sigma = 1$

5.22.

$$c = \sum_{r \in \mathcal{R}} (r \mod p) \tag{5.22}$$

**RDG1NamedCategoric**

The same data generator as for RDG1NamedContinuous but this time generating boolean attributes instead of numeric ones so now the rules are boolean predicates.

### 5.3.3 Results

In this section the results of the above described experiments are presented. Six plots are presented in Fig. 5.7. To clearly understand what the axes represent some notation has to be introduced. Let $\mathcal{R} = r_1, r_2, \ldots, r_n$ be the set relevant attributes and $\mathcal{I} = i_1, i_2, \ldots, i_m$ the set of irrelevant ones having $|\mathcal{R}| = n$ and $|\mathcal{I}| = m$. And let $w(a)$ be the weight assigned by the algorithm to attribute $a$. Now, the x-axis represents the total number of attributes $(m + n)$ and the y-axis the separability $s$ (i.e. the maximum weight assigned to a relevant attribute minus the maximum weight assigned to an irrelevant one). Formulas are shown in Eq. 5.23.

$$\begin{aligned} &\text{x-axis: } m + n \\ &\text{y-axis: } s = \left( \max_{a_r \in \mathcal{R}} w(a_r) \right) - \left( \max_{a_i \in \mathcal{I}} w(a_i) \right) \end{aligned} \tag{5.23}$$

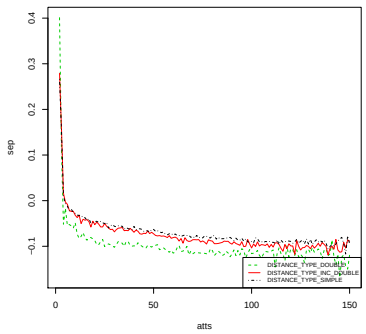
Now, in order to accentuate the global differences between the three algorithms six more plots are presented with the accumulated results for the y-axis. Fig. 5.8 shows these results. Now the x-axis keeps the same definition as before while the y-axis is the accumulated value of the separability, so now the formula for the y-axis value at point $x_n$ is the one in Eq. 5.24 knowing that $s_i$ is the separability defined in Eq. 5.23 at point $x_i$.

$$\text{y-axis: } \sum_{i=0}^{n} s_i \tag{5.24}$$

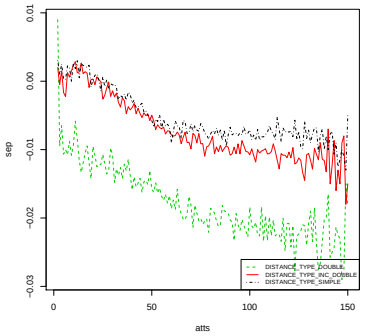
For this new axis definition, the slope of the function indicates positive or negative separability. If function descends at some point then separability was negative, on the other hand if function is ascending at this point then separability was positive. The steepness of the slope indicates the magnitude of the separability (either if it was positive or negative). And finally the separation between the curves for each algorithm tells about the accumulated difference of separabilities. If at the end one algorithm is above another it shows that the accumulated (and so the mean) separability is greater for this particular algorithm so one can conclude that in average this algorithm outperforms the other.

### 5.3.4 Conclusions

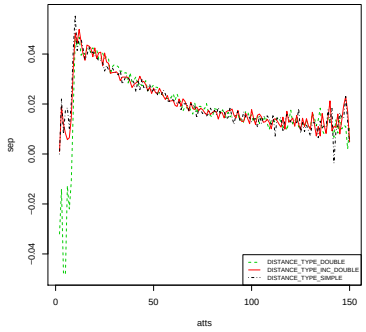
By looking at the results above, it can be seen that none of the three algorithms is clearly better than another for the chosen set of problems. Looking at the first set of plots having separability is in the x-axis, we can see that the curves for three algorithms are almost the same, only when there are few attributes DRELIEFF seems to have different behavior.



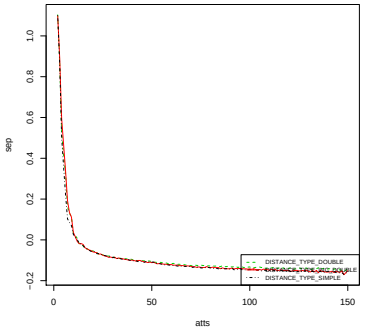

**(a)** RDG1NamedContinuous



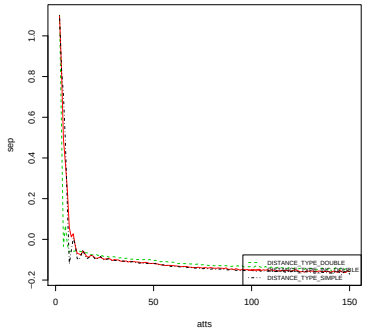

**(b)** RandomRBFRandRed



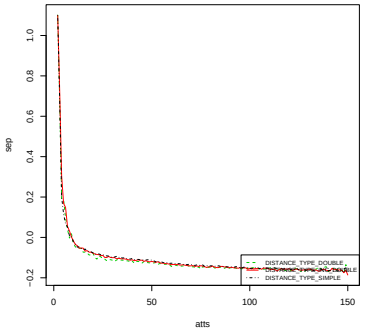

**(c)** NonMonotonic




**(d)** MajorityN

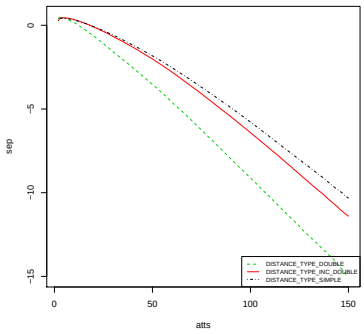



**(e)** ModuloP

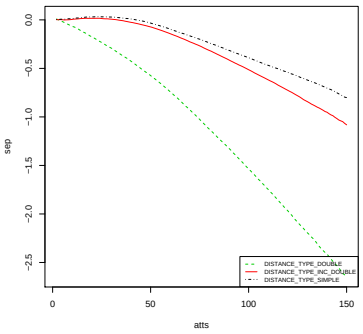



**(f)** RDG1NamedCategorical

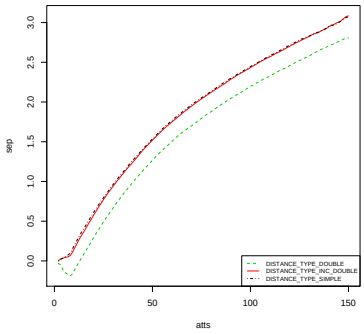
**Figure 5.7:** Separability versus total number of attributes for the three algorithms.



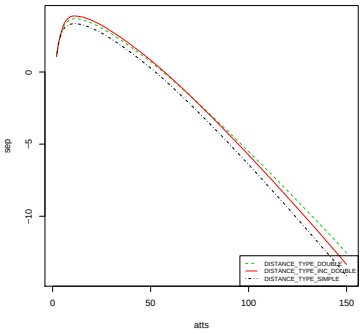

**(a)** RDG1NamedContinuous



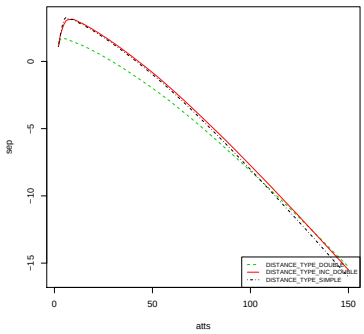

**(b)** RandomRBFRandRed



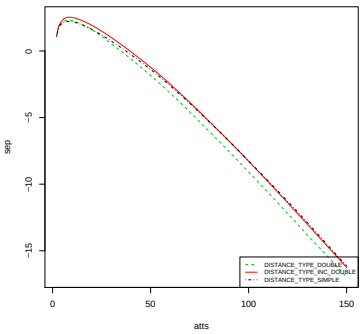

**(c)** NonMonotonic




**(d)** MajorityN




**(e)** ModuloP




**(f)** RDG1NamedCategorical

**Figure 5.8:** Accumulated separability versus total number of attributes for the three algorithms.

An anomaly is the problem of the random RBFs, there dReliefF is clearly worse. In fatct, except for the majority problem dReliefF is always the worse algorithm and even there it is non-significantly better. A difference between dReliefF and the other two algorithms is that it uses the calculated weights as distance ponderations starting at the first iteration of the algorithm. That certainly may cause ReliefF to get stuck into a local minimum found in those first iterations because the distance function that is using does not take into account some of the relevant variables. In a section above where pdReliefF is introduced, we stated the hypothesis that using the weights estimates since the first iteration may cause decrease performance due to the fact that these estimations may be too biased to the first instances and, so, may be far from the optimal weights. Now the results help support this hypothesis. That could also explain why dReliefF's behaviour is different from the others when few attributes are evaluated as opposed as when more attributes are present. When there are few attributes to calculate distance with, making a mistake on choosing their ponderations makes big changes in the results, so problems with few attributes are more sensible to wrong distance calculations and cause dReliefF to either have much higher or lower performance depending on how close are the early weights to the real optimal weights. If the first instances seen by the algorithm are not representative of the whole set, for example because they share some common characteristic that is rare among other instances, then the weights used will be biased; on the other hand if these first instances give more accurate weight approximates, then is possible that dReliefF's worked better than the rest.

There is also another characteristic of the results to be pointed out. In the second set of plots where differences among the algorithms stand out clearer, one can see differences between the behavior of the normal version of the algorithm as opposed to the modified ones. In these plots, two parallel curves for the separability of two algorithms, indicate that their performance evolves in the same way, meanwhile divergent curves indicate that the performance of one of them increases (decreases) more than the other. Having this in mind the results show that for the two first problems with numeric attributes the performance of dReliefF decreases very quick, normal ReliefF is the best of the three and pdReliefF is close to it though its performance also decreases faster than normal ReliefF's. Results for NonMonotonic are not clear as separability for that particular problem keeps very high for any number of attributes and the three algorithms perform almost identical. Some modifications could be applied to the generation of the problem to make it more difficult for ReliefF to discriminate attributes' relevance (e.g. adding more noise to the relevant ones) and compare the performance degradation for the three algorithms. The odd thing is that on the contrary of what happens with numeric problems, when we move onto the categoric ones we can see that now the algorithm which suffers the least performance decrease is dReliefF followed by pdReliefF.

The final conclusion looking at these experimental results must be that although the performance of the three algorithms is frequently almost the same, the new algorithm pdReliefF introduced seems to be always in the middle of the other two quite stick to the better of the two while the other two are better or worse depending on the problem type, maybe depending on whether attributes are numeric or categoric. And also that dReliefF is very sensible to early errors on weight approximation of ReliefF so it must be used carefully.

As future work, more problems could be tested and specific experiments should be conducted to get deeper in the hypothesis that the different versions of ReliefF perform different

on problems with numeric or categoric attributes. Also some tests on real data should be done using different classifiers to contrast them to the results on artificially generated ones.

Chapter

# 6 The Remainder Set of Features

In this chapter we explore one of the sources of instability we mentioned in the introduction: random parts of the algorithm. Specifically we tackle how sequential feature subset selection algorithms evaluate features. We describe a modification of these family of algorithms to test the hypothesis that the way these algorithms select features is a source of instability. If our hypothesis was right modification should lead to more stable results and maybe to better prediction power.

## 6.1 The Remainder Set of Features

As the goal of feature selection is to find an optimal subset $X^*$ as seen in Eq. (2.4), it seems plausible to choose an $X_k$ for each iteration as in (6.1) in a stepwise and greedy way, which is exactly what the previously described feature selection algorithms do:

$$X_k = \arg\max_{X \in \mathbf{S}_k} J(X), \qquad k = 1, \ldots, n \tag{6.1}$$

In real problems, features are far from independent, thus not always the best feature set in every iteration has to be the best option. Quite possibly there is some combination of features that would be a better choice than the feature which maximizes $J(X)$ in this iteration. So we see that the forward steps in the previous algorithms are not taking into account some information they could use. Only the *usefulness* of every generated subset of features is measured, as in (6.1). However, by considering the current set of features $X_k$ another set is implicitly created, the set of *remaining* features or *remainder set* $Y_k = Y \setminus X_k$. This set can also give information about the new variable to be added or removed at every step. It is our conjecture that a way to enhance the detection of feature interactions is to see how the addition of a feature to $X_k$ (a removal, from the point of view of $Y_k$) affects the *usefulness* of the remainder set.

The intuitive explanation is that the optimal set $X^*$ the algorithm is trying to find could be either in $X_k$, in $Y_k$ or split among the two. The evaluation criterion should give higher values to a set containing $X^*$ and the performance of this set should be affected when removing a feature from $X^*$. So, the idea is to add the most useful feature to $X_k$ and whose removal is most harmful to $Y_k$, i.e. to maximize $J(X_k)$ and minimize $J(Y_k)$. The general idea is called *Remainder Subset Awareness* for obvious reasons.

This idea tries to improve the weaknesses of SFG and SBG described below:

1. SFG at its first steps evaluates the features on their own, not taking into account the relationships between them [25], so two features that are very good when used together but that are not that good individually may not be selected.
2. SBG at its first steps evaluates each feature with all the irrelevant and redundant features, which may discard a *useful* feature due to the effects of the *unuseful* ones over the evaluation criterion.

In SFG we know for sure that initially $X^* \subseteq Y_0$. So looking at the features in $Y_0$ could be seen as some kind of in the search space lookahead without really looking at future states. By knowing that a feature removal is very harmful for $Y_k$ we know that even though this feature is not very good for $X_k$ right now it may be good for it in the future when other variables from $Y_k$ have been added to $X_k$. In the same way, in SBG if a feature about to be removed from $X_k$ behaves very well in $Y_k$ it may indicate that the interactions with the rest of features in $X_k$ are masking the true value of this feature and that this feature may be useful in the future when some other features have been discarded.

In a more formal manner, we know that by definition of $X^*$ in (2.4) we know that:

$$J(X^*) > J(X^* \setminus \{x\}), \qquad \forall x \in X^* \tag{6.2}$$

From the above equation, it should also be true (but it is not) that:

$$J(X_k \setminus \{z\}) > J(X_k \setminus \{x\}), \qquad \forall X_k \supsetneq X^*, \forall x \in X^*, \forall z \notin X^* \tag{6.3}$$

This equation states that removing a feature in $X^*$ from any set $X_k$ that contains $X^*$ is always more harmful than removing a feature not in $X^*$ from this same set.

If (6.3) was always true, SBG would always find $X^*$ as it would remove one feature not in $X^*$ at each step until $X^*$ was found. But it would only be always true if the $J$ criterion was not affected by the addition of *unuseful* features. It will not be true if the features in $X_k \setminus X^*$ affect the results of the $J$ criterion. As said in the introduction, irrelevant or redundant features may lead classifiers to find false regularities and learn from that instead of learning from the features that really determine the instance class. So, for example, $J$ may be higher due to overfitting on the input data.

Looking at the remainder can help bypassing these evaluation criteria limitations. Two artificial problems have been chosen to illustrate the benefits of the remainder set awareness. As the best solution to these problems is known the benefits of the new algorithm can clearly be explained. These problems have been chosen because they have some special characteristics that make either SFG or SBG fail to find the best solution.

These two problems are:

**corrAl** This dataset has two classes and six boolean features ($A_0$; $A_1$; $B_0$; $B_1$; $I$; $C$). Feature $I$ is irrelevant, feature $C$ is correlated to the class label 75% of the time, and the other four features are relevant to the boolean target concept: $(A_0 \wedge A_1) \vee (B_0 \wedge B_1)$. SFG will choose $C$ first as it is the best feature when taken all alone [34]. The hypothesis is that the *usefulness* of the remainder set would be so high if $C$ was chosen that the modified version of SFG would not choose it.

**antiCorrAl** This dataset has been generated ad hoc for this chapter. It is a three class problem with 11 continuous features $(I_1, I_2, \ldots, I_9, C_1, C_2)$. The class is numeric and can be 1, 2 o 3. Features $I_1$ to $I_9$ are random values of a normal distribution with mean equal to the class of the example and standard deviation of 1. So the value of the feature $n$ for the example $i$ that has class $Y_i$, is generated as $I_{ni} = rnorm(\mu = Y_i, \sigma = 1)$, where $rnorm$ is a function that generates random deviates for the normal distribution. Feature $C_1$ is generated as $C_{1i} = rnorm(\mu = Y_i, \sigma = 0.5)$. Finally the last feature $C_2$ is generated by the formula: $C_{2i} = C_{1i} - Y_i + rnorm(\mu = 1, \sigma = 0.2)$. So the problem is separable and using $C_1$ and $C_2$ the class can be easily predicted. SBG will discard $C_2$ the first as $I$ features mask the *usefulness* of the $C$ group and $C_2$ is the worst feature when taken all alone. But if we had to choose the feature which most harmed $Y_0$, it would be $C_1$, so the hypothesis is that the remainder set aware would choose $C_1$ in the first place and then as the $C$ group is the best, it would chose $C_2$ finding the best solution.

The experiments were run using the algorithm and experimental setup explained in the following sections (this setup includes an external loop of cross-validation so the feature subset selection was executed 10 times for each dataset).

**SFG on corrAl**: The hypothesis was confirmed: a conventional SFG chose feature $C$ the first in most cases followed by the other features. Surprisingly the final solution contained the irrelevant feature on some cases. On the other hand the modified version almost always chose one of the relevant features in the first place and the final solution was the best one most of the time. Sometimes the feature $C$ was chosen in the third place as adding one of the relevant features to the current set din't make it much better (two were missing) and removing it from the remainder set was not that bad (the other two were missing there). That is the worst scenario where $X^*$ is completely split between $X_k$ and $Y_k$.

**SBG on antiCorrAl**: The hypothesis was also confirmed. The conventional SBG discarded $C_2$ in most cases. The median of the iteration number where it discarded $C_2$ was 3,5 and it found the best solution in some cases. On the other hand the remainder set aware SFG always selected $C_1$ and $C_2$. In most of the runs the best solution was found.

Table 6.1 shows the mean error rates for the SFG or SBG and for the remainder set aware SFG (RSA) and the p-value of the Wilcoxon-Mann-Whitney test showing that the difference on the means is statistically significant. It also shows the median number of selected features for each algorithm with its absolute deviation.

## 6.2 Combination function

With the above formulation we have a multi-objective problem, since not always the subset with maximum $J(X_k)$ will be the same as the subset with minimum $J(Y_k)$. So it will not

**Table 6.1:** Results for the SFG on corrAl and SBG on antiCorrAl datasets

| Problem | $\mu$err | $\mu$err RSA | p-val | #feat | #feat RSA |
|---|---|---|---|---|---|
| corrAl | 0.077 | 0.009 | **0.002** | $5.00 \pm 0.00$ | 4.00±0.00 |
| antiCorrAl | 0.132 | 0.023 | **0.007** | 7.00±2.22 | 2.00±0.00 |

be possible to satisfy both objectives with the same single solution. In this case, either the two solutions have to be explored or a trade-off has to be found that partly optimizes both objectives. If both solutions are chosen for further exploration, then the search space is highly increased over the original version of the algorithm, and the complexity of the algorithm grows from polynomial to exponential, which is unfeasible. A reasonable alternative is to choose the subset which maximizes some predefined function $f$ of the two criteria among the two candidate subsets, as expressed by:

$$\underset{X \in \mathbf{S}_k}{\arg\max}\ f\left[J(X), J(Y \setminus X)\right], \qquad k = 1, \ldots, n \tag{6.4}$$

The function $f : (0,1)^2 \rightarrow (0,1)$ has to be chosen to be continuous in both arguments, increasing in the first and decreasing in the second and to permit control on the relative importance of the two arguments (thus it is non-symmetrical). Following this alternative, an algorithm of the sequential kind can be modified by replacing the evaluation function $J(X)$ with the one in (6.4). As an example, **Algorithm 6.1** shows the straightforward

**Algorithm 6.1:** Remainder set aware SFG (RSA)

1: $X_0 \leftarrow \emptyset$ {Initial subset}
2: $i \leftarrow 0$
3: **repeat**
4: $\quad \mathbf{S}_{i+1} \leftarrow \{X \mid X = X_i \cup \{x\} \wedge x \in Y \setminus X_i\}$ {Subset generation}
5: $\quad X_{i+1} \leftarrow \underset{X \in \mathbf{S}_{i+1}}{\arg\max}\, f\left[J(X), J(Y \setminus X)\right]$ {Subset evaluation}
6: $\quad i \leftarrow i + 1$
7: **until** $J(X_i) \leq J(X_{i-1}) \vee i = n$ {Stopping criterion}
8: **return** $X_{i-1}$ {Selected subset}

*Remainder Subset Aware* version of the original SFG presented in **Algorithm 2.1**. Other forward/backward algorithms would be modified analogously though we will only consider the forward version in this work.

The chosen evaluation function $f$, which combines the *usefulness* of the selected subset of features with that of the remaining subset is shown in (6.5).

$$f(x, y) = \frac{x * wx - y * wy + 1}{2},\ wx, wy \in (0, 1) \tag{6.5}$$

Note that $wx = 1 \wedge wy = 0$ recovers the conventional algorithms and $wx = 0.5 \wedge xy = 0.5$ corresponds to mean between $x$ and $1 - y$. In general, greater values of $wy$ over $wx$ give more weight to the evaluation of the inducer in the remainder set. The values of these weights have been selected taking into account the weaknesses of SFG and SBG presented on the previous section. As seen before on one hand a set of irrelevant features may hide a good variable and on the other hand a bad feature when taken all alone could improve when evaluated in a group. So it is not the size of the set that matters. We chose the weights to be proportional to the *usefulness* of the set we are about to modify. So, if we wanted to compute $f[J(X_k), J(Y/X_k)]$, the weights would be $wx = J(X_{k-1})$ and $wy = J(Y/X_{k-1})$. This setting gives more importance to the better sets of features. So, when $X_k$ is better than $Y_k$, the features that make it even better are prefered. But when $Y_k$ is better than $X_k$ (e.g. at the first steps of SFG) the features that harm $Y_k$ the most are prefered over others that helped more $X_k$.

## 6.3 Experimental work

Experimental work is presented in order to assess the described modification with a group of four sequential algorithms, using the datasets described in Section 3.1.

The algorithms were implemented using the R language for statistical computing [61] in order to implement conventional SFG and SBG algorithms and the modified remainder set aware version of SFG. The experimental setup consists of the two nested cross-validation loops described in 3.2. For every fold and repetition of the outer cross-validation loop, a two feature selection processes are conducted with the same examples, i.e. one with the original algorithm and one with the RSA. Both SFG and SBG are compared with the remainder set aware version of SFG. Each feature selection iteration uses a learner and another 5x2-fold cross-validation for estimating feature *usefulness*. In our case we have run the experiments using three different learners: the 1-nearest neighbor (1NN) [80] (which uses Euclidean distance), the Fisher's linear discriminant analysis (LDA) [21] and a support vector machine (SVM) [12] with a linear kernel (the regularization constant or cost and the kernel width) are kept fixed to their default values in all the experiments, since we are only interested in the influence that different feature subsets have on the modelling[1]. It is important to mention that there was no stopping criterion in the experiments: forward methods run until all the features were selected and backward ones until all of them were removed. Then the best of the obtained sequence of subsets was returned. Once the best subset of features is found a test is conducted on the features that did not participate in the feature selection process using the same 1NN algorithm. Then the classification error is returned and a Wilcoxon-Mann-Whitney test is made on the resulting set of classification errors from the two algorithms to determine if the difference is statistically significant. The results are displayed in Tables 6.2 and 6.3. The tables also show the median of the size of the final selected subsets and its absolute deviation.

[1]These values are 1 for the cost parameter and the inverse of the number of features for the smoothing parameter in the kernel.

### 6.3.1 Stability results

As the main objective of our work is to improve stability of FSS algorithms an assessment of the stability for each execution has been performed. In addition to the above displayed classification error we computed the stability of the FSS algorithm results using Kuncheva's index. Since the stability index needs a constant number of selected features we calculated the stability for each size of the possible features sets instead of doing so for only the sizes of the best sets. Tables 6.4 and 6.5 show the summary stability results by averaging all the stability scores for each subset size. Appendix A contains all the charts showing the individual stability for each possible subset size.

## 6.4 Discussion

A few datasets have shown statistically significant results according to the Wilcoxon-Mann-Whitney test. In all the statistically significant differences RSA is better than the original SFG or SBG algorithms.

**Table 6.2:** Classification error and number of features comparing SFG to RSA. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Err | $\text{Err}_{RSA}$ | p-val | NF | $\text{NF}_{RSA}$ |
|---|---|---|---|---|---|---|
| diabetes | knn | 0.314 | 0.321 | 0.238 | 3.6 | 4.2 |
| heart-statlog | knn | 0.239 | **0.282** | 0.02 | 5 | 6.3 |
| ionosphere | knn | 0.113 | **0.13** | 0.007 | 7.2 | 6.6 |
| landsat_train | knn | 0.118 | 0.112 | 0.11 | 20.1 | 21.6 |
| leaf | knn | 0.398 | **0.419** | 0.04 | 9.5 | 9.5 |
| ma_breast_cancer | knn | 0.303 | 0.289 | 0.453 | 35.1 | 46.1 |
| ma_colon_tumor | knn | 0.239 | 0.232 | 0.416 | 25.1 | 54.4 |
| ma_gcm | knn | 0.482 | 0.459 | 0.116 | 49.7 | 61 |
| ma_leukemia | knn | 0.094 | 0.092 | 0.383 | 2.5 | 5 |
| ma_lung_cancer | knn | 0.03 | 0.033 | 0.444 | 2 | 6.7 |
| ma_prostate_cancer | knn | 0.112 | 0.107 | 0.361 | 30.4 | 33.8 |
| mammogram | knn | 0.3 | 0.288 | 0.419 | 18 | 18.3 |
| parkinsons | knn | 0.152 | 0.171 | 0.071 | 9.7 | 10.9 |
| pop_failures | knn | 0.084 | 0.081 | 0.222 | 6.6 | 5.8 |
| sonar | knn | 0.212 | 0.186 | 0.11 | 19.8 | 25.8 |
| spectf | knn | 0.253 | 0.247 | 0.323 | 10.3 | 9.4 |
| vehicle | knn | 0.312 | **0.326** | 0.026 | 11.3 | 9.6 |
| waveform | knn | 0.218 | 0.221 | 0.161 | 15.9 | 16.3 |
| wdbc | knn | 0.083 | 0.092 | 0.223 | 16 | 15.4 |
| diabetes | lda | 0.238 | **0.234** | 0.032 | 4.7 | 4.9 |
| heart-statlog | lda | 0.178 | 0.17 | 0.389 | 6.9 | 6.8 |
| landsat_train | lda | 0.161 | 0.164 | 0.138 | 17.3 | 17.6 |
| leaf | lda | 0.25 | 0.242 | 0.203 | 10.3 | 11 |
| ma_breast_cancer | lda | 0.318 | 0.342 | 0.297 | 14.1 | 16.4 |
| ma_colon_tumor | lda | 0.232 | 0.242 | 0.444 | 18.7 | 65.8 |
| ma_gcm | lda | 0.502 | 0.485 | 0.203 | 47.3 | 72.6 |
| ma_leukemia | lda | 0.094 | 0.09 | 0.453 | 11.2 | 32.5 |
| ma_lung_cancer | lda | 0.033 | 0.032 | 0.399 | 3.2 | 4.7 |
| ma_prostate_cancer | lda | 0.262 | 0.251 | 0.306 | 19.7 | 50 |
| mammogram | lda | 0.138 | 0.158 | 0.136 | 6.5 | 6 |
| pop_failures | lda | 0.057 | 0.061 | 0.187 | 8.5 | 10.4 |
| spectf | lda | 0.232 | 0.224 | 0.221 | 7.8 | 9.3 |
| vehicle | lda | 0.24 | 0.235 | 0.277 | 15.2 | 14.5 |
| waveform | lda | 0.146 | 0.145 | 0.078 | 18.3 | 17.6 |
| wdbc | lda | 0.043 | 0.04 | 0.156 | 13.3 | 12.8 |
| diabetes | svm | 0.24 | 0.235 | 0.339 | 4.9 | 5.1 |
| heart-statlog | svm | 0.187 | 0.174 | 0.13 | 7 | 6.8 |
| ionosphere | svm | 0.142 | 0.146 | 0.176 | 10.8 | 8.7 |
| landsat_train | svm | 0.135 | 0.137 | 0.092 | 18.3 | 20.5 |
| leaf | svm | 0.278 | 0.271 | 0.078 | 11.2 | 11.5 |

Continued on next page

**Table 6.2 – continued from previous page**

| Problem | Ind. | Err | $\text{Err}_{RSA}$ | p-val | NF | $\text{NF}_{RSA}$ |
|---|---|---|---|---|---|---|
| ma_breast_cancer | svm | 0.27 | 0.289 | 0.312 | 39.5 | 25.4 |
| ma_colon_tumor | svm | 0.203 | 0.177 | 0.102 | 23.3 | 38 |
| ma_leukemia | svm | 0.089 | 0.061 | 0.143 | 3.5 | 10.6 |
| ma_lung_cancer | svm | 0.03 | 0.03 | 0.444 | 2.2 | 3.4 |
| ma_prostate_cancer | svm | 0.191 | 0.166 | 0.07 | 40.6 | 52 |
| mammogram | svm | 0.195 | 0.155 | 0.118 | 7.8 | 13.2 |
| parkinsons | svm | 0.15 | 0.154 | 0.201 | 6.9 | 7.7 |
| pop_failures | svm | 0.06 | 0.058 | 0.658 | 8.3 | 11.5 |
| sonar | svm | 0.276 | 0.288 | 0.107 | 12.1 | 19.3 |
| spectf | svm | 0.233 | 0.231 | 0.5 | 13.9 | 14.6 |
| vehicle | svm | 0.22 | 0.227 | 0.096 | 15.7 | 15.2 |
| waveform | svm | 0.137 | 0.135 | 0.239 | 17.4 | 17.9 |
| wdbc | svm | 0.03 | 0.033 | 0.337 | 10 | 10.1 |
| **Average** | | 0.1933 | 0.1919 | | 14.6075 | 19.0774 |

**Table 6.3:** Classification error and number of features comparing SBG to RSA. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Err | $\text{Err}_{RSA}$ | p-val | NF | $\text{NF}_{RSA}$ |
|---|---|---|---|---|---|---|
| diabetes | knn | 0.32 | 0.325 | 0.337 | 5.1 | 4.3 |
| heart-statlog | knn | 0.255 | 0.211 | 0.07 | 3.7 | 3.4 |
| ionosphere | knn | 0.119 | 0.121 | 0.541 | 8 | 10.7 |
| landsat_train | knn | 0.115 | 0.111 | 0.154 | 25.9 | 30.6 |
| leaf | knn | 0.408 | 0.418 | 0.11 | 8.5 | 9.9 |
| ma_breast_cancer | knn | 0.276 | 0.247 | 0.077 | 42.5 | 51.7 |
| ma_colon_tumor | knn | 0.21 | 0.223 | 0.249 | 32.6 | 35.9 |
| ma_gcm | knn | 0.573 | **0.533** | 0.003 | 41.4 | 29.8 |
| ma_leukemia | knn | 0.122 | **0.083** | 0.029 | 12.1 | 13.7 |
| ma_lung_cancer | knn | 0.027 | 0.021 | 0.211 | 33.2 | 30.8 |
| ma_prostate_cancer | knn | 0.157 | 0.171 | 0.361 | 39.9 | 16.1 |
| mammogram | knn | 0.278 | 0.3 | 0.317 | 10.9 | 16.3 |
| parkinsons | knn | 0.136 | 0.151 | 0.089 | 12.9 | 12.7 |
| pop_failures | knn | 0.095 | 0.091 | 0.238 | 8.6 | 7.4 |
| sonar | knn | 0.233 | 0.21 | 0.277 | 20.9 | 15 |
| spectf | knn | 0.268 | 0.265 | 0.571 | 11.5 | 9.5 |
| vehicle | knn | 0.317 | 0.329 | 0.181 | 12.1 | 9.6 |
| waveform | knn | 0.224 | 0.221 | 0.193 | 16.8 | 17.1 |
| wdbc | knn | 0.078 | 0.088 | 0.071 | 15.5 | 23 |

Continued on next page

**Table 6.3 – continued from previous page**

| Problem | Ind. | Err | $\text{Err}_{RSA}$ | p-val | NF | $\text{NF}_{RSA}$ |
|---|---|---|---|---|---|---|
| diabetes | lda | 0.235 | 0.235 | 0.383 | 4.7 | 5.7 |
| landsat_train | lda | 0.164 | **0.161** | 0.026 | 17.5 | 32.1 |
| leaf | lda | 0.245 | 0.247 | 0.453 | 11.9 | 11.8 |
| ma_breast_cancer | lda | 0.322 | 0.272 | 0.179 | 29.1 | 30.1 |
| ma_colon_tumor | lda | 0.19 | 0.232 | 0.219 | 63 | 55.8 |
| ma_gcm | lda | 0.496 | 0.477 | 0.22 | 117 | 119.1 |
| ma_leukemia | lda | 0.139 | 0.136 | 0.639 | 41.8 | 22.2 |
| ma_lung_cancer | lda | 0.039 | 0.039 | 0.682 | 26.9 | 23.4 |
| ma_prostate_cancer | lda | 0.271 | 0.312 | 0.053 | 32.1 | 58.6 |
| mammogram | lda | 0.179 | 0.212 | 0.361 | 6.4 | 10.6 |
| pop_failures | lda | 0.055 | **0.059** | 0.029 | 8.1 | 7.8 |
| sonar | lda | 0.278 | 0.265 | 0.278 | 14.1 | 9.5 |
| spectf | lda | 0.235 | 0.231 | 0.287 | 8.7 | 8 |
| vehicle | lda | 0.239 | **0.226** | 0.029 | 13.8 | 16 |
| waveform | lda | 0.145 | 0.145 | 0.287 | 17.9 | 18.1 |
| wdbc | lda | 0.043 | 0.042 | 0.406 | 9.2 | 12.4 |
| diabetes | svm | 0.238 | 0.24 | 0.36 | 5.2 | 5.5 |
| heart-statlog | svm | 0.183 | 0.169 | 0.186 | 7.5 | 7 |
| ionosphere | svm | 0.145 | 0.136 | 0.061 | 10.6 | 8.8 |
| landsat_train | svm | 0.138 | **0.132** | 0.012 | 17.4 | 33.7 |
| leaf | svm | 0.267 | 0.274 | 0.187 | 11.6 | 11.3 |
| ma_breast_cancer | svm | 0.266 | 0.251 | 0.187 | 23.1 | 21.4 |
| ma_colon_tumor | svm | 0.213 | 0.2 | 0.316 | 18 | 28 |
| ma_gcm | svm | 0.417 | 0.409 | 0.361 | 63.9 | 47 |
| ma_leukemia | svm | 0.083 | 0.084 | 0.472 | 9.2 | 5.5 |
| ma_lung_cancer | svm | 0.033 | 0.029 | 0.361 | 7.7 | 5.9 |
| ma_prostate_cancer | svm | 0.203 | **0.148** | 0.007 | 31.7 | 54.8 |
| mammogram | svm | 0.175 | 0.165 | 0.305 | 8 | 8 |
| parkinsons | svm | 0.147 | 0.143 | 0.13 | 8.3 | 8.1 |
| pop_failures | svm | 0.056 | 0.054 | 0.296 | 7.3 | 6.7 |
| sonar | svm | 0.278 | 0.262 | 0.203 | 11.1 | 11.7 |
| spectf | svm | 0.24 | 0.236 | 0.287 | 9.4 | 8.5 |
| vehicle | svm | 0.227 | 0.221 | 0.207 | 14.9 | 16 |
| waveform | svm | 0.137 | 0.136 | 0.461 | 18.6 | 17.6 |
| wdbc | svm | 0.039 | **0.031** | 0.006 | 12.7 | 16.6 |
| **Average** | | 0.2031 | 0.1986 | | 20.0093 | 20.5704 |

**Table 6.4:** Stability results comparing SFG to RSA. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Stab | $\text{Stab}_{RSA}$ | p-val |
|---|---|---|---|---|
| diabetes | knn | 0.316±0.1128 | 0.3913±0.1414 | 0.4688 |
| heart-statlog | knn | **0.4524±0.2383** | 0.234±0.1416 | 0.001 |
| ionosphere | knn | 0.1316±0.0954 | 0.102±0.0527 | 0.0898 |
| landsat_train | knn | 0.2434±0.2069 | 0.2168±0.0982 | 0.942 |
| leaf | knn | 0.2322±0.0983 | **0.5389±0.1845** | 0.0017 |
| ma_breast_cancer | knn | **0.0666±0.0296** | 0.037±0.025 | 0 |
| ma_colon_tumor | knn | **0.1273±0.0552** | 0.0307±0.0201 | 0 |
| ma_gcm | knn | 0.2716±0.115 | **0.294±0.1505** | 0 |
| ma_leukemia | knn | **0.2908±0.1166** | 0.0234±0.0321 | 0 |
| ma_lung_cancer | knn | **0.3359±0.1361** | 0.0459±0.0228 | 0 |
| ma_prostate_cancer | knn | **0.1087±0.063** | 0.0548±0.0275 | 0 |
| mammogram | knn | 0.0768±0.0401 | 0.0712±0.0232 | 0.5699 |
| parkinsons | knn | **0.2952±0.1668** | 0.0509±0.0623 | 0 |
| pop_failures | knn | 0.2176±0.1359 | **0.2942±0.1647** | 2e-04 |
| sonar | knn | **0.1871±0.0746** | 0.0976±0.0674 | 0 |
| spectf | knn | **0.0438±0.0309** | 0.0265±0.0345 | 0.0187 |
| vehicle | knn | **0.6687±0.151** | 0.4173±0.0856 | 0 |
| waveform | knn | 0.4067±0.1407 | **0.6078±0.1777** | 2e-04 |
| wdbc | knn | 0.1736±0.2352 | 0.2336±0.2475 | 0.4622 |
| diabetes | lda | 0.507±0.3332 | 0.6192±0.2613 | 0.2945 |
| heart-statlog | lda | 0.1323±0.1181 | **0.3969±0.1022** | 5e-04 |
| landsat_train | lda | 0.1864±0.2149 | **0.2148±0.0983** | 0.0074 |
| leaf | lda | 0.1697±0.0977 | **0.3887±0.134** | 0.0024 |
| ma_breast_cancer | lda | -0.0202±0.0144 | **-0.0097±0.012** | 0 |
| ma_colon_tumor | lda | -0.0162±0.0149 | -0.0182±0.0069 | 0.4136 |
| ma_gcm | lda | **0.0127±0.0263** | 0.0028±0.0288 | 0 |
| ma_leukemia | lda | **0.1182±0.0799** | 0.0034±0.0155 | 0 |
| ma_lung_cancer | lda | **0.0097±0.0169** | 0.001±0.0093 | 0 |
| ma_prostate_cancer | lda | -3e-04±0.0195 | **0.0109±0.0249** | 0 |
| mammogram | lda | 0.0915±0.1304 | 0.0889±0.1139 | 0.8655 |
| pop_failures | lda | 0.1995±0.1363 | 0.2371±0.1525 | 0.0525 |
| spectf | lda | **-0.0209±0.0373** | -0.0454±0.0132 | 0 |
| vehicle | lda | 0.3202±0.2164 | 0.3281±0.1968 | 1 |
| waveform | lda | 0.4546±0.112 | **0.5929±0.1914** | 0.0083 |
| wdbc | lda | 0.1097±0.119 | 0.1142±0.0473 | 0.1004 |
| diabetes | svm | 0.4266±0.341 | 0.4446±0.3683 | 0.5896 |
| heart-statlog | svm | 0.2333±0.1556 | 0.2619±0.0949 | 0.083 |
| ionosphere | svm | 0.0979±0.1661 | **0.178±0.1135** | 3e-04 |
| landsat_train | svm | **0.22±0.2442** | 0.1378±0.0892 | 0.0126 |
| leaf | svm | 0.1649±0.0877 | **0.4612±0.2178** | 0.0012 |

Continued on next page

**Table 6.4 – continued from previous page**

| Problem | Ind. | Stab | $\text{Stab}_{RSA}$ | p-val |
|---|---|---|---|---|
| ma_breast_cancer | svm | **-0.0055±0.0132** | -0.0084±0.0105 | 0.0033 |
| ma_colon_tumor | svm | **-0.0051±0.0169** | -0.019±0.0082 | 0 |
| ma_leukemia | svm | **0.3708±0.1013** | -0.0087±0.0076 | 0 |
| ma_lung_cancer | svm | **0.4847±0.1387** | 0.0065±0.0112 | 0 |
| ma_prostate_cancer | svm | **0.0272±0.0253** | 0.0227±0.0329 | 0 |
| mammogram | svm | 0.035±0.0903 | **0.1028±0.1062** | 0 |
| parkinsons | svm | 0.0705±0.1106 | 0.0574±0.0609 | 0.5168 |
| pop_failures | svm | **0.1532±0.144** | 0.0506±0.1194 | 0.0067 |
| sonar | svm | 0.0171±0.0754 | **0.0335±0.0293** | 3e-04 |
| spectf | svm | **0.1356±0.2313** | -0.0145±0.0224 | 0 |
| vehicle | svm | 0.1732±0.0924 | **0.3294±0.1704** | 0.0099 |
| waveform | svm | 0.5186±0.118 | 0.5994±0.1951 | 0.1327 |
| wdbc | svm | 0.1319±0.1003 | 0.1687±0.0606 | 0.0667 |
| **Average** | | 0.1915±0.1166 | 0.1792±0.0922 | |

**Table 6.5:** Stability results comparing SBG to RSA. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Stab | $\text{Stab}_{RSA}$ | p-val |
|---|---|---|---|---|
| diabetes | knn | 0.4751±0.2878 | 0.5647±0.2722 | 0.0754 |
| heart-statlog | knn | 0.1846±0.2314 | **0.3806±0.1782** | 0.0067 |
| ionosphere | knn | 0.1313±0.1236 | **0.2456±0.1612** | 0 |
| landsat_train | knn | 0.1252±0.1949 | **0.4437±0.1982** | 0 |
| leaf | knn | 0.2824±0.1673 | **0.5586±0.1574** | 0.0011 |
| ma_breast_cancer | knn | 0.0154±0.0692 | **0.0339±0.0761** | 0 |
| ma_colon_tumor | knn | 0.0068±0.0705 | **0.0983±0.0807** | 0 |
| ma_gcm | knn | 0.0123±0.0792 | **0.1073±0.0707** | 0 |
| ma_leukemia | knn | 0.1133±0.0742 | 0.1113±0.0747 | 0.1145 |
| ma_lung_cancer | knn | 0.0345±0.0626 | **0.1741±0.0858** | 0 |
| ma_prostate_cancer | knn | 0.0218±0.0821 | **0.0992±0.1017** | 0 |
| mammogram | knn | 0.1175±0.173 | **0.149±0.1683** | 0 |
| parkinsons | knn | 0.1659±0.2134 | **0.256±0.2477** | 5e-04 |
| pop_failures | knn | 0.3325±0.2825 | **0.36±0.2849** | 0.0294 |
| sonar | knn | 0.0614±0.1145 | **0.2851±0.1262** | 0 |
| spectf | knn | 0.0652±0.1266 | **0.089±0.1324** | 0 |
| vehicle | knn | 0.5209±0.1585 | 0.4759±0.166 | 0.4079 |
| waveform | knn | 0.4604±0.1009 | **0.5916±0.1881** | 0.002 |
| wdbc | knn | 0.2159±0.2315 | **0.3366±0.2925** | 0 |

Continued on next page

**Table 6.5 – continued from previous page**

| Problem | Ind. | Stab | $\text{Stab}_{RSA}$ | p-val |
|---|---|---|---|---|
| diabetes | lda | 0.567±0.3934 | 0.5883±0.3707 | 0.2807 |
| landsat_train | lda | 0.1576±0.1929 | 0.1601±0.1915 | 0.6318 |
| leaf | lda | 0.3087±0.2012 | **0.4404±0.2861** | 0.01 |
| ma_breast_cancer | lda | 0.0152±0.0652 | 0.0157±0.0675 | 0.1233 |
| ma_colon_tumor | lda | -0.005±0.0671 | **0.0546±0.0645** | 0 |
| ma_gcm | lda | 0.017±0.0809 | **0.0358±0.0862** | 0 |
| ma_leukemia | lda | **0.1015±0.0582** | 0.0313±0.0582 | 0 |
| ma_lung_cancer | lda | **0.0579±0.056** | 0.011±0.0583 | 0 |
| ma_prostate_cancer | lda | 0.0139±0.0831 | **0.0357±0.075** | 0 |
| mammogram | lda | **0.1171±0.1742** | 0.1012±0.162 | 0 |
| pop_failures | lda | **0.4186±0.356** | 0.3416±0.3075 | 2e-04 |
| sonar | lda | 0.0601±0.1192 | **0.2019±0.1435** | 0 |
| spectf | lda | **0.0836±0.1319** | 0.0556±0.1279 | 0 |
| vehicle | lda | **0.4471±0.2554** | 0.2433±0.1385 | 3e-04 |
| waveform | lda | 0.4013±0.175 | **0.5926±0.1645** | 1e-04 |
| wdbc | lda | 0.0904±0.1624 | **0.3056±0.1541** | 0 |
| diabetes | svm | 0.5457±0.3915 | 0.54±0.378 | 0.7874 |
| heart-statlog | svm | 0.313±0.234 | **0.4994±0.2561** | 0.0017 |
| ionosphere | svm | 0.1653±0.1953 | **0.3009±0.2301** | 0 |
| landsat_train | svm | 0.1605±0.2554 | **0.2632±0.223** | 0 |
| leaf | svm | 0.3486±0.1165 | **0.4941±0.1985** | 0.0057 |
| ma_breast_cancer | svm | 0.0186±0.0717 | **0.0349±0.0719** | 0 |
| ma_colon_tumor | svm | 0.0056±0.0638 | **0.0627±0.0617** | 0 |
| ma_gcm | svm | 0.0157±0.0814 | **0.0677±0.0828** | 0 |
| ma_leukemia | svm | **0.4118±0.073** | 0.0454±0.0566 | 0 |
| ma_lung_cancer | svm | **0.5206±0.2155** | 0.0521±0.0549 | 0 |
| ma_prostate_cancer | svm | 0.0184±0.081 | **0.101±0.0919** | 0 |
| mammogram | svm | 0.1324±0.2 | **0.1783±0.2026** | 0 |
| parkinsons | svm | 0.1361±0.1832 | **0.2184±0.2282** | 0.0094 |
| pop_failures | svm | 0.4104±0.2957 | 0.3778±0.2867 | 0.0673 |
| sonar | svm | 0.038±0.1217 | **0.142±0.108** | 0 |
| spectf | svm | **0.0942±0.1233** | 0.0676±0.1313 | 0 |
| vehicle | svm | **0.3793±0.1979** | 0.2365±0.1415 | 7e-04 |
| waveform | svm | 0.4776±0.1266 | 0.5411±0.1548 | 0.348 |
| wdbc | svm | 0.1107±0.1783 | **0.3576±0.1544** | 0 |
| **Average** | | 0.1944±0.1597 | 0.2436±0.1599 | |

It is seen that for both SFG performance is in general increased (as expressed by the chosen $J$) at the price of selecting some more features from an average of 14.6 for the non modified version to 19 for the RSA. In the case of the backward version, the number of selected features is almost exactly the same in both versions and also the performance roughly the same on average but we find a greater number of problems where the performance is significantly better. Whenever the conventional and the modified algorithm are in ties or very close to, the modified versions offer a solution with a lower number of features, which is also interesting from the point of view of feature selection. Detailed experiment results are displayed in Tables 6.2 and 6.3. Regarding stability we can see very different results between SFG and SBG and among the different inducers. Even though the results for stability of SFG are greater 65% of the time only 33% of them are statistically significant, very similar to the percentage of executions, 30%, where stability was worsened. Taking a deeper look at the results we can see that they vary a lot among inducers. While SBG 1NN achieved the best results, SFG 1NN was the most harmed one while for the LDA and SVM learners SFG is highly benefited from taking the remainder set of features into account. In the case of LDA, only one problem has lower stability results when using the RSA version and 40% of them have statistically significant better results. Globally 35 of the total 54 executions of SBG have significantly higher stability and only 19% have a significantly lower value. The results for SBG with the 1NN learner are the most remarkable: we have made them more stable in 84% of the problems and for no problem they are worse.

## 6.5 Conclusions

We have presented a modification for feature subset selection algorithms that iteratively evaluate subsets of features, by making them compute not only the *usefulness* of the selected set but also the *usefulness* of the remainder set. A set of experiments have been conducted in order to compare the modified versions of the algorithms with their original versions. Our experimental results indicate a very significant stability improvement and in some cases improvement in performance too while keeping the size of the final subset roughly equal or lower. The fact that the modified version does not always improve the results of the original should not be a surprise. According to the *No free lunch* theorems , if an algorithm achieves superior results on some problems, it must pay with inferiority on other problems. However, it is possible to modify a search algorithm to obtain a version that is generally superior in performance to the original version [84]. In the present situation this fact can be explained by the way the modified version selects subsets of features. For instance, given two features: One that makes a significant reduction of the performance of the remainder set and not a big change on the performance of the selected set. And one that increases the performance of the selected set a bit more than the first one but does not make a big change on the remainder one. A conventional algorithm would always select the latter while the modified version would maybe select the former. That could lead the modified version to avoid local maxima by not selecting the best feature in this iteration feature and end with a better subset; but when the algorithm has selected a set close to optimal subset, the modification may cause the algorithm to loose precision in choosing features. The improved results on stability can intuitively be explained by the way the algorithm is using information about features. It is reevaluating features that have been discarded in previous SBG iterations (or not yet included for SFG) thus somehow evaluating the whole feature set at each iteration. This gives it a better overview

of the features and makes it more robust to dataset perturbations. An interesting fact are the differences among the different inducers that should be further investigated in future work.

Chapter

# 7 Exploiting the Accumulated Evidence

This chapter studies another of the hypothesis presented in Chapter 1: the intrinsic instability introduced by using a inducer in *wrapper* sequential feature subset selection algorithms. The result of the inducer, which is a random variable, is used to assess feature importance. We propose to diminish the effect of this random variable by *remembering* all of the previous evaluations of each feature. If our hypothesis is true, by accumulating the results of different evaluations we should obtain more stable results.

## 7.1 Introduction

The selection of a new feature (either to be removed or added to the current set) involves the evaluation of many models. These models typically consist of the addition (deletion) of one feature to (from) the current set. As we have seen in Chapter 2 *wrapper* methods use and inducer to build temporary solutions and return their evaluation using some resampling method (e.g. cross-validation) [39].

In the standard procedure, only the *best* such model evaluation is considered for selecting which feature should removed or added, and the remaining evaluations are readily *discarded*. Yet there *is* valuable information in the discarded evaluations: the very many evaluated subsets contain information on the relevance of the features that belong to the subset; this relevance does not depend on the subset being selected or not. When an inducer is requested to estimate the predictive accuracy of a model using a given feature subset within a wrapper strategy, no indication is given on which feature is the most recent addition (or deletion): the inducer just sees a feature subset which has to be evaluated *as a whole*.

Since the most difficult part of a FSS process is to evaluate the *interactions* between features, the *accumulated* evaluation of a feature in diverse contexts should account for many of these interactions, and ultimately provide with a more informed estimation of usefulness for the chosen inducer. The different *contexts* of a particular feature $x$ are given by all those

subsets which are being evaluated along the search process (not necessarily to assess the influence of $x$, as noted above), either containing or *not* containing $x$.

Our idea is to accumulate the inducer evaluations as a rich source of information. This information can then be used in conventional existing algorithms, such as the well-known *forward* or *backward* selection. This idea can be applied to any sequential search algorithm and any inducer and, as shown below, at a negligible extra cost.

Here we present experimental results showing good performance in a suite of benchmark microarray problems. The proposed modification always achieves improvements when applied to standard backward selection, either in the estimated predictive accuracy, in the size of the delivered gene subsets, or in both.

## 7.2 Accumulated evidence and feature relevance

The idea consists on accumulating the *evidence* in favor or against a feature, taking into account its *history* of evaluations alongside different feature subsets. A further explanation can be to extract the most of every subset evaluation, normally the most costly part of a FSS process.

Let $Y_x = \{X \in \mathcal{P}(Y) | x \in X\}$ be the set of all feature subsets of the initial set that contain a certain feature $x$ (note that $|Y_x| = 2^{n-1}$ for all $x \in Y$).

Let $\mathcal{L}_x^+$ and $\mathcal{L}_x^-$ be the average evaluation of all subsets containing and *not* containing $x$:

$$\mathcal{L}_x^+ = \frac{1}{2^{n-1}} \sum_{X \in Y_x} J_{\mathcal{L}}(X)$$

$$\mathcal{L}_x^- = \frac{1}{2^{n-1}} \sum_{X \notin Y_x} J_{\mathcal{L}}(X)$$

Given an inducer $\mathcal{L}$ (either filter or wrapper) define, for a given feature $x \in Y$, the *relevance* of $x$ as:

$$R_{\mathcal{L}}(x) = \mathcal{L}_x^+ - \mathcal{L}_x^- \tag{7.1}$$

The above definition can be more compactly expressed as:

$$R_{\mathcal{L}}(x) = \frac{1}{2^{n-1}} \sum_{X \notin Y_x} \Big( J_{\mathcal{L}}(X \cup \{x\}) - J_{\mathcal{L}}(X) \Big) \tag{7.2}$$

**Remark 1**. Defining feature relevance with expression (7.2) is very attractive, since it captures feature interactions in all possible ways. We take the freedom of presenting an informal but hopefully illustrative analogy of what this measure captures. Imagine we are willing to evaluate the average influence of a *basketball* player on a team scoring: we can compute the difference in points that the team scores *with* and *without* this player, no matter

what other players are playing in the player's team. If this difference is positive, then we can conclude that this player's accomplishments are positive for the team; otherwise we conclude that we should better sell the player at the best possible price! Note that in this example, only subsets $X$ of size 4 are considered and $Y \setminus X$ is the bench[1].

**Remark 2**. Full evaluation of expression (7.2) has an exponential cost in $n$, making it unfeasible for most practical applications; an estimation is therefore mandatory via Monte Carlo techniques, generating feature subsets randomly from a precise probability distribution determined by the FSS algorithm being used. Oddly, although $R_{\mathcal{L}}(x)$ takes into account all possible feature interactions, by its very nature it does not capture redundancy: two identical features will have the same relevance. This is true even by making $J_{\mathcal{L}}$ cope with redundancy. However, since a search algorithm will impose an order on the evaluated feature subsets, the current state can be used to ascertain redundancy, as will be shown below.

The above expressions can be conveniently generalized by considering a *weighing* function $w$:

$$R^w_{\mathcal{L}}(x) = \frac{\sum\limits_{X \not\in Y_x} \Big( J_{\mathcal{L}}(X \cup \{x\}) - J_{\mathcal{L}}(x) \Big)\ w_x(X)}{\sum\limits_{X \not\in Y_x} w_x(X)} \tag{7.3}$$

For example, the choice $w_x(X) = |X|/|Y| = |X|/n$ gives more importance to improvements in $J_{\mathcal{L}}$ achieved in a scenario with already many features (improving performance in such a case has a certain merit); alternatively, one could choose $w_x(X) = J_{\mathcal{L}}(X)$; this choice expresses the belief that an improved performance when $J_{\mathcal{L}}(X)$ is already high should be rewarded, and less so when it is low (it has a much lower merit). Many alternatives are possible and the best one (if such choice exists at all) is at the moment an open question. Note that eq. (7.3) reduces to eq. (7.1) when $w_x(X) = 1$ for all $x$.

In the following, we present a practical method to approximate this measure of relevance and integrate it in a SBG search algorithm at no additional cost. The idea consists on accumulating the *evidence* in favor or against a feature by taking into account the *history* of evaluations throughout the search process.

### 7.2.1 Practical computation of the accumulated evidence

Let $X_k$ denote the current set, where $|X_k| = k$, for notational simplicity (thus $X_0 = \emptyset$ and $X_n = Y$); let $X_{n-k}$ be the set of features not in $X_k$, i.e. $X_{n-k} = Y \setminus X_k$. Assume first we are in front of performing a *forward* step. Given $X_k$, in a classical SFG, the set

$$\Big\{ J_{\mathcal{L}}(X_k \cup \{x\}) \mid x \in X_{n-k} \Big\} \text{ is computed} \tag{7.4}$$

and the feature $x' = \arg\max\limits_{x \in X_{n-k}} J_{\mathcal{L}}(X_k \cup \{x\})$ is selected. However, all the remaining information:

[1] Incidentally, this way of ranking players (together with rebounds, assists, etc) is used in the NBA.

$$\left\{J_{\mathcal{L}}(X_k \cup \{x\}) \mid x \in X_{n-k}, x \neq x'\right\} \text{ is discarded,} \tag{7.5}$$

yet sometime in the future these individual features $x$ (and eventually $x'$ itself) will be considered again for inclusion or exclusion from the current set in forward or backward steps, respectively.

Conversely, in a *backward step* the search algorithm is going to evaluate a feature $x$ for possible exclusion from $X_{n-k}$ in such a way that the set

$$\left\{J_{\mathcal{L}}(X_{n-k} \setminus \{x\}) \mid x \in X_{n-k}\right\} \text{ is computed} \tag{7.6}$$

and the feature $x' = \arg\max_{x \in X_{n-k}} J_{\mathcal{L}}(X_{n-k} \setminus \{x\})$ is selected for removal. Again, the information:

$$\left\{J_{\mathcal{L}}(X_{n-k} \setminus \{x\}) \mid x \in X_{n-k}, x \neq x'\right\} \text{ is discarded.} \tag{7.7}$$

Yet, sometime in the future these individual features $x$ (and eventually $x'$ itself) will be considered again for inclusion or exclusion from the current set in forward or backward steps, respectively. Reasoning in more general terms, the search algorithm always evaluates a feature $x$ for possible inclusion in (or exclusion from) the current subset using information about $x$.

Now let $P_{\mathcal{L}}$ denote the set of feature subsets that the search algorithm has evaluated so far (implying a call to $\mathcal{L}$). Let $P_{\mathcal{L}|x} = \{X \in P_{\mathcal{L}} | x \in X\}$. For every $x \in Y$, define the accumulated evaluations (or simply *accumulators*) as the Monte Carlo estimations:

$$\hat{\mathcal{L}}_x^+ = \frac{\sum_{X \in P_{\mathcal{L}|x}} J_{\mathcal{L}}(X) w_x(X)}{\sum_{X \in P_{\mathcal{L}|x}} w_x(X)} \tag{7.8}$$

$$\hat{\mathcal{L}}_x^- = \frac{\sum_{X \notin P_{\mathcal{L}|x}} J_{\mathcal{L}}(X) w_x(X)}{\sum_{X \notin P_{\mathcal{L}|x}} w_x(X)} \tag{7.9}$$

which are approximations to the weighted versions of $\mathcal{L}_x^+$ and $\mathcal{L}_x^-$, respectively. These two approximated values depend on the search algorithm, which determines the strategy to traverse the search space. Different FSS algorithms (such as SFG or SBG) provide different traces of evaluated subsets at any given number of algorithmic steps. In these conditions, the impact of the considered feature in the *current subset* $X$ can be used to ascertain redundancy and make it influence the search, by modullating the effect of the accumulated evaluations. Consider now, for $\lambda \in [0, 1]$,

$$\hat{R}_{\mathcal{L}}^w(x) = \frac{\lambda}{2}(\hat{\mathcal{L}}_x^+ - \hat{\mathcal{L}}_x^- + 1) + (1 - \lambda)\hat{J}_{\mathcal{L}}(x), \tag{7.10}$$

where $\hat{J}_{\mathcal{L}}(x) = J_{\mathcal{L}}(X \setminus \{x\})$ in a backward step (the effect of removing $x$ from $X$) and $\hat{J}_{\mathcal{L}}(x) = J_{\mathcal{L}}(X \cup \{x\})$ in a forward step (the effect of adding $x$ to $X$) and $\lambda$ is a free parameter. This scheme generalizes conventional forward and backward steps (as used by SFG, SBG or any other sequential algorithm) in two ways:

1. By setting $\lambda = 0$, the conventional forward and backward steps are recovered and both relevance and redundancy are evaluated using $\hat{J}_{\mathcal{L}}(x)$. By setting $\lambda = 1$, a pure arithmetic average between $\hat{\mathcal{L}}_x^+$ and $1 - \hat{\mathcal{L}}_x^-$ is computed.

   For other values of $\lambda$, the *search history* makes an influence on the search itself, conditioning the selection of features. In this case, only a $1 - \lambda$ fraction of the importance is assigned to the current subset evaluation.

2. The search history itself is formed by all known contexts in which the considered feature could appear or not (and not only by previous evaluations of the feature), thus conforming a broader picture of its true relevance.

**Example**. Consider the following feature subset mask ($n = 20$) for a current feature subset $X_8 \subset Y$ where the $i$-th index is 1 when feature $x_i \in X_8$ and 0 otherwise:

```
10010010001010100101
```

signaling the presence of features number $1, 4, 7$, etc. An evaluation $J_{\mathcal{L}}(X)$ of this subset is indeed expressing how good is to have the first feature but not the second or the third, also how good is to have the seventh feature but not the one before the last, and so forth. For this reason, all the features in $Y$ (and not only those in $X$) should have their accumulators updated every time.

## 7.3 A practical algorithm

We illustrate the approach on the popular SBG search algorithm (**Algorithm 2.2**) and give a practical implementation of the previous ideas for it (SBG$^+$, **Algorithm 7.1**). In addition, for simplicity of presentation, we fix $w_x(X) = 1$. In this case, normalization simply amounts to a division by the number of performed accumulations. The initialization of the accumulated relevances is 0 for all $x \in Y$. The results are first accumulated and then used; for this reason, even in the first algorithmic step (the first discarded feature) the behavior of both algorithms may start to diverge. At the end of the FSS process, $n_x^+$ (resp. $n_x^-$) will be the number of times that a feature subset (resp. not) containing $x$ has been evaluated. Note that the computation is done at a negligible overhead in cost; this is due to the fact that the inducer is called *exactly* the same number of times for SBG than for the accumulated counterpart SBG$^+$.

**Algorithm 7.1:** SBG$^+$ (inducer $\mathcal{L}$, feature set $Y$, $\lambda \in [0,1]$)

$X_n \leftarrow Y$
$k \leftarrow 0$
{Initialize accumulators and counters}
$\forall x \in Y, \hat{\mathcal{L}}_x^+ \leftarrow \hat{\mathcal{L}}_x^- \leftarrow 0$
$\forall x \in Y, n_x^+ \leftarrow n_x^- \leftarrow 0$
**repeat**
  **for all** $x \in X_{n-k}$ **do**
    compute the set $\left\{ J_{\mathcal{L}}(X_{n-k} \setminus \{x\}) \right\}$
  **end for**
  {Update accumulators and counters}
  **for all** $x \in Y$ **do**
    **if** $x \in X_{n-k}$ **then**
      $\hat{\mathcal{L}}_x^+ \leftarrow \hat{\mathcal{L}}_x^+ + \sum_{y \in X_{n-k} \setminus \{x\}} J_{\mathcal{L}}(X_{n-k} \setminus \{y\})$
      $n_x^+ \leftarrow n_x^+ + 1$
    **else**
      $\hat{\mathcal{L}}_x^- \leftarrow \hat{\mathcal{L}}_x^- + J_{\mathcal{L}}(X_{n-k} \setminus \{x\})$
      $n_x^- \leftarrow n_x^- + 1$
    **end if**
  **end for**
  $x' \leftarrow \arg\max_{x \in X_{n-k}} \left\{ \frac{\lambda}{2}(\hat{\mathcal{L}}_x^+ / n_x^+ - \hat{\mathcal{L}}_x^- / n_x^- + 1) + (1-\lambda)\hat{J}_{\mathcal{L}}(X_{n-k} \setminus \{x\}) \right\}$
  $X_{n-k} \leftarrow X_{n-k} \setminus \{x'\}$
  $k \leftarrow k + 1$
**until** $k = n$
**return** $\arg\max_{k=1 \div n} J_{\mathcal{L}}(X_k)$

## 7.4 Experimental work

Experimental work is now presented in order to assess the described modifications using two sequential algorithms: SBG and its accumulated counterpart SBG$^+$. The algorithms were implemented using the R language for statistical computing [61].

The experimental setup consists of the two nested cross-validation loops described in 3.2. For every fold and repetition of the outer cross-validation loop, two feature selection processes are conducted with the same examples, one with the original algorithm (SBG) and one with the accumulated version (SBG$^+$).

Each feature selection iteration uses the 1-*nearest-neighbor* learner implementation in [80] (which uses Euclidean distance), *linear discriminant analysis* (LDA) and the *Support Vector Machine* with radial kernel (SVM$_r$). The parameters of the SVM (here again the regularization constant or cost and the kernel width) are kept fixed to their default values in all the

experiments, since we are only interested in the influence that different feature subsets have on the modelling[2].

The evaluation of these inducers is resampled in a second (*inner*) 5x2cv loop for a more informed estimation of usefulness. In all cases, stratification is used to keep the same proportion of class labels across the partitioned sets. After some preliminary experiments, we set $\lambda = \frac{2}{3}$ in expression (7.10). It is very important to mention that there is no stopping criterion in the algorithms: the two backward methods run until all the features have been removed. Then the *best* subset in the obtained sequence of subsets is returned. This setting avoids the specification of an a priori size for the solution. It also eliminates the possibility that the accumulated algorithm performs differently simply because it merely influences the stopping point.

Once the best feature subset is found (a different one in every outer loop), this subset is evaluated in the corresponding test set. The final test error (the one reported) is the mean of these 10 values.

### 7.4.1 Benchmarking microarray data sets

In a microarray gene expression context, there is a wide spectrum of FSS algorithms. Commonly found methods fall into the *filter* category: a list of the top-ranked genes based on some inducer-free figure of merit is generated, followed by and inductive process where a classifier is incrementally evaluated [64]. This constitutes a fast and low complexity approach. However, considering individual contributions only can hinder the discovery of possible interactions between genes.

Many authors have claimed that the *wrapper* approach, if affordable, is preferable to the filter approach (e.g. [47, 39] ). It is therefore of the greatest importance to take the most of every evaluation of the inducer, which is normally the more costly part.

The description of the used datasets can be found at Section 3.1. It is important to stress that there has been little effort to find the best models among those represented by the considered inducers: in other words, nearest-neighbors is limited to just one neighbour and the SVM parameters have been set to their default values. All the effort is devoted to find good feature subsets and to compare the two search algorithms in similar experimental circumstances.

### 7.4.2 Stability results

Here again stability for the modified versions will be assessed by using Kuncheva's index for each size of the possible features sets. Table 7.2 show the summary stability results by averaging all the stability scores for each subset size. Appendix B contains all the charts showing the individual stability for each possible subset size.

[2]These values are 1 for the cost parameter and the inverse of the number of features for the smoothing parameter in the kernel.

**Table 7.1:** Classification error and number of features comparing $SBG^+$ to $SBG$. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Err | Err$^+$ | p-val | NF | NF$^+$ |
|---|---|---|---|---|---|---|
| diabetes | knn | 0.314 | 0.315 | 0.571 | 4.5 | 4.8 |
| heart-statlog | knn | 0.227 | 0.236 | 0.429 | 4.4 | 3.9 |
| ionosphere | knn | 0.136 | 0.132 | 0.238 | 7.2 | 7.2 |
| landsat_train | knn | 0.113 | **0.116** | 0.042 | 23.1 | 23.3 |
| leaf | knn | 0.424 | 0.438 | 0.22 | 9.5 | 9.2 |
| ma_breast_cancer | knn | 0.293 | 0.262 | 0.054 | 34.2 | 60.2 |
| ma_colon_tumor | knn | 0.2 | 0.181 | 0.053 | 73.8 | 37.4 |
| ma_gcm | knn | 0.552 | 0.548 | 0.249 | 49.2 | 44.2 |
| ma_leukemia | knn | 0.108 | 0.081 | 0.088 | 28.3 | 7.2 |
| ma_lung_cancer | knn | 0.034 | 0.033 | 0.5 | 20 | 17.4 |
| ma_prostate_cancer | knn | 0.154 | 0.14 | 0.141 | 28.3 | 18.3 |
| mammogram | knn | 0.272 | 0.279 | 0.367 | 17.6 | 17.3 |
| parkinsons | knn | 0.142 | 0.142 | 0.706 | 9.2 | 13.6 |
| pop_failures | knn | 0.091 | 0.092 | 0.318 | 6.2 | 5.8 |
| sonar | knn | 0.205 | 0.193 | 0.342 | 19.9 | 25.3 |
| spectf | knn | 0.282 | 0.287 | 0.4 | 11.5 | 16.1 |
| vehicle | knn | 0.325 | 0.315 | 0.143 | 11.6 | 10.5 |
| waveform | knn | 0.222 | 0.221 | 0.342 | 16.6 | 16.2 |
| wdbc | knn | 0.083 | 0.084 | 0.446 | 18.5 | 19.2 |
| antiCorrAl | lda | 0.005 | 0.004 | 0.5 | 2.9 | 3 |
| diabetes | lda | 0.234 | 0.239 | 0.062 | 5.3 | 5.7 |
| heart-statlog | lda | 0.174 | **0.184** | 0.028 | 6.8 | 7.4 |
| iris | lda | 0.071 | 0.064 | 0.231 | 2 | 2.2 |
| landsat_train | lda | 0.163 | 0.163 | 0.406 | 15.8 | 18 |
| leaf | lda | 0.241 | 0.225 | 0.078 | 10.9 | 11.2 |
| ma_breast_cancer | lda | 0.368 | **0.274** | 0.014 | 52.6 | 22.4 |
| ma_colon_tumor | lda | 0.223 | 0.19 | 0.174 | 79.2 | 70.5 |
| ma_gcm | lda | 0.478 | 0.46 | 0.385 | 136.2 | 147.2 |
| ma_leukemia | lda | 0.157 | 0.167 | 0.406 | 32.5 | 30 |
| ma_lung_cancer | lda | 0.033 | 0.027 | 0.567 | 13.4 | 4.1 |
| ma_prostate_cancer | lda | 0.265 | 0.248 | 0.297 | 44.3 | 23.5 |
| mammogram | lda | 0.184 | 0.158 | 0.136 | 5.5 | 5.8 |
| pop_failures | lda | 0.056 | 0.055 | 0.228 | 7.9 | 7.5 |
| sonar | lda | 0.281 | 0.279 | 0.5 | 9.2 | 11.8 |
| spectf | lda | 0.219 | 0.231 | 0.092 | 9.3 | 8 |
| vehicle | lda | 0.233 | 0.236 | 0.239 | 15.9 | 14.2 |
| waveform | lda | 0.144 | **0.147** | 0.033 | 17.8 | 16.4 |
| wdbc | lda | 0.047 | 0.045 | 0.383 | 11.2 | 12.9 |
| diabetes | svm | 0.238 | 0.236 | 0.241 | 5.2 | 4.7 |
| heart-statlog | svm | 0.183 | 0.186 | 0.476 | 7.5 | 7.7 |

Continued on next page

**Table 7.1 – continued from previous page**

| Problem | Ind. | Err | Err$^+$ | p-val | NF | NF$^+$ |
|---|---|---|---|---|---|---|
| ionosphere | svm | 0.145 | 0.141 | 0.399 | 10.6 | 8.7 |
| landsat_train | svm | 0.138 | **0.135** | 0.042 | 17.4 | 20.9 |
| leaf | svm | 0.267 | 0.265 | 0.476 | 11.6 | 11.5 |
| ma_breast_cancer | svm | 0.256 | 0.237 | 0.22 | 17.5 | 13 |
| ma_colon_tumor | svm | 0.187 | 0.181 | 0.304 | 14.2 | 15.5 |
| ma_gcm | svm | 0.527 | 0.512 | 0.203 | 16.6 | 18.6 |
| ma_leukemia | svm | 0.092 | 0.078 | 0.439 | 37.2 | 6.1 |
| ma_lung_cancer | svm | 0.033 | 0.034 | 0.312 | 8.8 | 4.5 |
| ma_prostate_cancer | svm | 0.22 | 0.219 | 0.361 | 8.1 | 12.9 |
| mammogram | svm | 0.362 | 0.453 | 0.088 | 43.3 | 39.8 |
| parkinsons | svm | 0.147 | 0.14 | 0.238 | 8.3 | 5.8 |
| pop_failures | svm | 0.056 | 0.055 | 0.417 | 7.3 | 6.7 |
| sonar | svm | 0.278 | 0.266 | 0.318 | 11.1 | 18.3 |
| spectf | svm | 0.24 | 0.228 | 0.13 | 9.4 | 9.9 |
| vehicle | svm | 0.227 | 0.229 | 0.444 | 14.9 | 13.7 |
| waveform | svm | 0.137 | 0.138 | 0.203 | 18.6 | 17.5 |
| wdbc | svm | 0.039 | 0.035 | 0.171 | 12.7 | 9.7 |
| **Average** | | 0.2022 | 0.1976 | | 20.2211 | 17.9719 |

**Table 7.2:** Stability results comparing $SBG^+$ to $SBG$. Figures in boldface correspond to statistically significant improvements.

| Problem | Ind. | Stab | Stab$^+$ | p-val |
|---|---|---|---|---|
| diabetes | knn | 0.4427±0.238 | 0.4699±0.3412 | 0.726 |
| heart-statlog | knn | 0.3022±0.1885 | 0.3053±0.1679 | 0.8888 |
| ionosphere | knn | 0.1581±0.1412 | 0.16±0.2081 | 0.9047 |
| landsat_train | knn | 0.188±0.1667 | 0.1902±0.1838 | 0.9187 |
| leaf | knn | 0.2745±0.1829 | **0.3586±0.1486** | 0.0025 |
| ma_breast_cancer | knn | 0.0107±0.0644 | **0.0253±0.0738** | 0 |
| ma_colon_tumor | knn | 0.0061±0.0689 | **0.0254±0.0737** | 0 |
| ma_gcm | knn | 0.0086±0.0727 | **0.015±0.0834** | 0.0053 |
| ma_leukemia | knn | **0.0701±0.0687** | 0.0396±0.0814 | 0 |
| ma_lung_cancer | knn | **0.0328±0.0644** | 0.014±0.077 | 0 |
| ma_prostate_cancer | knn | 0.0331±0.0837 | **0.0455±0.0984** | 0 |
| mammogram | knn | **0.0952±0.1582** | 0.0824±0.1866 | 0.0232 |
| parkinsons | knn | 0.2129±0.2082 | 0.2015±0.1867 | 0.5315 |
| pop_failures | knn | 0.3704±0.2675 | 0.3566±0.2917 | 0.1182 |
| sonar | knn | 0.0674±0.1146 | **0.083±0.1113** | 2e-04 |

Continued on next page

**Table 7.2 – continued from previous page**

| Problem | Ind. | Stab | Stab$^+$ | p-val |
|---|---|---|---|---|
| spectf | knn | 0.0629±0.1191 | **0.0946±0.1196** | 0 |
| vehicle | knn | 0.4685±0.0851 | **0.5434±0.0813** | 0.0013 |
| waveform | knn | 0.448±0.1382 | 0.4607±0.1233 | 0.3048 |
| wdbc | knn | 0.2603±0.2426 | **0.3286±0.2563** | 0 |
| antiCorrAl | lda | 0.6006±0.3146 | 0.545±0.3287 | 0.1073 |
| diabetes | lda | 0.5139±0.4005 | 0.58±0.3317 | 0.2945 |
| heart-statlog | lda | 0.3238±0.2264 | 0.33±0.28 | 0.8888 |
| iris | lda | 0.3181±0.2509 | 0.4292±0.2008 | 0.0975 |
| landsat_train | lda | **0.1375±0.2005** | 0.1253±0.2052 | 0.0276 |
| leaf | lda | 0.4454±0.2894 | 0.4282±0.2878 | 0.4324 |
| ma_breast_cancer | lda | -0.0011±0.0705 | **0.0151±0.0682** | 0 |
| ma_colon_tumor | lda | 0.0054±0.065 | **0.013±0.0703** | 0 |
| ma_gcm | lda | 0.0236±0.0824 | 0.0228±0.0885 | 0.1252 |
| ma_leukemia | lda | **0.2043±0.0765** | 0.0205±0.062 | 0 |
| ma_lung_cancer | lda | **0.0386±0.0595** | 0.0068±0.0667 | 0 |
| ma_prostate_cancer | lda | 0.0083±0.0683 | **0.0357±0.0735** | 0 |
| mammogram | lda | 0.0852±0.1942 | **0.1546±0.2164** | 0 |
| pop_failures | lda | 0.4219±0.3428 | 0.4205±0.3549 | 0.675 |
| sonar | lda | **0.0549±0.1217** | 0.0442±0.1119 | 0 |
| spectf | lda | **0.0744±0.1307** | 0.0599±0.1212 | 0.0048 |
| vehicle | lda | 0.4491±0.2655 | **0.521±0.2553** | 4e-04 |
| waveform | lda | 0.4315±0.1413 | **0.4718±0.1967** | 0.0181 |
| wdbc | lda | 0.0826±0.1623 | **0.1247±0.1446** | 0 |
| diabetes | svm | 0.5457±0.3915 | 0.5498±0.4046 | 1 |
| heart-statlog | svm | 0.313±0.234 | 0.2985±0.2364 | 0.7266 |
| ionosphere | svm | 0.1653±0.1953 | **0.1983±0.2196** | 0.0031 |
| landsat_train | svm | 0.1605±0.2554 | **0.2002±0.2517** | 0 |
| leaf | svm | 0.3486±0.1165 | 0.3593±0.1451 | 0.3794 |
| ma_breast_cancer | svm | 0.0155±0.0784 | 0.0165±0.0713 | 0.0589 |
| ma_colon_tumor | svm | **0.014±0.0672** | 0.0108±0.0694 | 0.0052 |
| ma_gcm | svm | 0.0148±0.085 | **0.0355±0.0899** | 0 |
| ma_leukemia | svm | **0.3589±0.1162** | 0.0112±0.0656 | 0 |
| ma_lung_cancer | svm | **0.4243±0.1778** | 0.0183±0.0582 | 0 |
| ma_prostate_cancer | svm | 0.0112±0.0864 | **0.0385±0.0966** | 0 |
| mammogram | svm | **0.1038±0.095** | 0.0332±0.0995 | 0 |
| parkinsons | svm | 0.1361±0.1832 | 0.1526±0.2158 | 0.2372 |
| pop_failures | svm | 0.4104±0.2957 | 0.4395±0.3218 | 0.3604 |
| sonar | svm | 0.038±0.1217 | **0.0626±0.1168** | 0 |
| spectf | svm | **0.0942±0.1233** | 0.0803±0.119 | 0.016 |
| vehicle | svm | 0.3793±0.1979 | **0.4493±0.2266** | 3e-04 |
| waveform | svm | 0.4776±0.1266 | **0.5162±0.1184** | 0.0458 |
| wdbc | svm | 0.1107±0.1783 | 0.1092±0.1533 | 0.537 |

Continued on next page

**Table 7.2 – continued from previous page**

| Problem | Ind. | Stab | Stab$^+$ | p-val |
|---|---|---|---|---|
| **Average** | | 0.2079±0.1625 | 0.2057±0.1656 | |

## 7.5 Discussion

The results of the FSS process are displayed in Table 7.1. The first table shows the (cross-validated) average test error for the two algorithms and the different inducers. The second table shows the (cross-validated) average size of the final selected subsets.

The first fact to note is that the accumulated version outperforms the standard version (though in general by a modest margin) in all cases. This is a very remarkable result, given the big differences among the problems and among the inducers. We also can see that the average classification error for the SBG$^+$ version among all problems and inducer algorithms is slightly lower (0.1976 versus 0.2022 of the non modified version). Second, SBG$^+$ finds in general solutions of lower size than SBG does, sometimes by a substantial amount (e.g., 1NN in *Colon Tumor* and *Leukemia*, most of LDA, or *Leukemia* and *Lung Cancer* with the SVM). Given that there is no stopping condition, our explanation is that the standard backward version is *greedier* than the accumulated one. By the (early) inclusion of some (or many) features that are not as good as they look in that moment, and cannot be removed, SBG is driven toward worse local minima of the error function as compared to SBG$^+$. The greediness itself is explained by the purely *local* (in the temporal sense) character of SBG and it also explains the worse prediction results of this algorithm.

Feature selection appears to be a viable avenue for dimensionality reduction in this field: a reduction of two orders of magnitude in the number of features by univariate methods shows substantial improvements (Table 3.1). With a further reduction of another order of magnitude, mean performance of the finally selected classifiers is similar to that achieved using the previously reduced subset. This behavior is important, both for computational and scientific reasons. Even without optimization of free parameters (a necessary step in normal conditions), cross-validated wrapper computations with 200 features may take several days of computing time on a modest machine. Scientifically, coping with hundreds of features and pretending interpretability of the role of every feature in the model is out of the question in many cases. This is aggravated in the present situation of data scarcity.

The results diverge for different classifiers, as it may be reasonably expected. This is of the greatest importance when assessing whether an improvement is consistent, or is limited to a certain type of method. In this sense, 1NN seems to be the best method for *Prostate Cancer*, LDA for *Lung Cancer* and the SVM for the other three (in all cases using SBG$^+$). The SVM tends to deliver smaller gene subsets, both for SBG and SBG$^+$. Given that the SVM parameters were not optimized beyond educated guesses, we think there is room for further improvement in the modeling, specially on the accuracy side.

Comparison to other results in the literature using the same data sets is a delicate undertaking in general. The methodological steps can be very different, especially concerning resampling techniques. We have found that many times there are no true test sets: feature subsets or model parameters (or both) are optimized by means of one or several resampled runs of cross-validation. This procedure is dangerous in that it cannot deliver an unbiased estimation of true error, given that, although test observations have not been used to create the model, they have been used to decide upon competing ones (namely, in the feature selection process itself). The stability of these results is also compromised if only one resample is carried out. On the other hand, the delivered gene subset size is a very important issue to bear in mind, if the solutions are to become interpretable and useful from the clinical point

of view. That said, we compare with several references illustrative of some work on the same data. A comprehensive list of recent usages of these datasets can be found in the review [5] conducted by Bolón-Canedo et al.:

1. For the *Colon Tumor* data set, [81] report an error of 12.7% with 94 genes, while [9] report an error of 23.0% with 33 genes, both using radial SVMs. For this dataset, we report a test error of 18.1% using an average of 15 genes.

2. For the *Leukemia* problem, the original poster [22] report a cross-validated median prediction strength of 0.77 and [9] report an error of 4.0% with 30 genes using a radial kernel, and an extraordinary 1.4% using only two genes and filter methods for ranking is reported by [30]. For this dataset, we report an average test error of 6.1% using an average of 6 genes.

3. The *Lung Cancer* data set is apparently the easiest to separate. Accuracy values as high as 99% are achieved by [9] (using a SVM and 38 genes) and by [32], this time using 5NN and as much as 135 genes. For this dataset, we report an average test error of 2.7% using an average of 4 genes.

4. In the *Prostate Cancer* problem, as low as 7% error as been reported (half our best result) using a radial SVM and 47 genes (nearly three times our result) [9].

5. For the *Breast Cancer* problem, an error of 21% is reported using a radial SVM and 46 genes [9], and an error of 32% using again a SVM and 8 genes [30]. For this dataset, we report an average test error of 23.7% using an average of 13 genes.

6. Finally for *GCM*, an error of 29,2% is reported using SVM one vs. all and 30 genes [62]. For this dataset, we report an average test error of 40.6% using an average of 147 genes.

Regarding stability we can see a variety of results. Even though the average results on stability are almost identical (0.2057 for $\text{SBG}^+$ versus 0.2079 for SBG), we can see a significant improve of the FSS stability in almost half of the problems (43%). For another 36% the difference in stability is not statistically significant. And 21% of them suffer from worse stability. When using 1NN learner we find the most stability improvement with 47% of the problems being more stable and only 16% being less stable. This learner also drew the least stable results among the three in their non-modified versions.

## 7.6 Conclusions

This chapter has presented a modification suitable for feature subset selection algorithms that iteratively evaluate subsets of features, by making them accumulate all the "log of merit" of the features in quite different contexts. The idea consists in that the current subset evaluation is not used directly to select the feature to add (or remove), but to *accumulate* information on the usefulness of the feature in many contexts. The different contexts of a particular feature $x$ are given by all those subsets that contain $x$ (they express how good is to have $x$) and do not contain $x$ (they express how good is not to have $x$). The accumulated information is then used to decide which feature should be added or removed (namely, that feature with the highest

(lowest) accumulated usefulness which has not yet been added (removed)). Therefore, the search history makes an influence on the search itself, conditioning the selection of features. This view is consistent with the definition of a search algorithm as a mapping from its history (including its present state) to the set of possible moves. In these conditions, less importance is assigned to the current subset evaluation than in a classical FSS setting (where it is the *only* source of information). Our experimental results indicate a general improvement in stability and performance, without any additional modelling effort. We have seen that the improvements on stability are greater when using learners that lead to less stable results when not using the accumulated information.

Future work may include exploring SFG. The decision to study SBG in the first place is consistent with the goal of discovering feature interactions. Having all the features from the beginning greatly facilitates this task. Nonetheless, the more modest computational demands that SFG entails in practice (if cut before exhaustion of features) may be an appealing characteristic. It is relevant to point out that the presented algorithmic modification may be of little help if an algorithm has many opportunities to rectify its decisions (*e.g.*, the PTA$(l, r)$ family of algorithms). However, even in this case, the forward or backward steps will be more informed, possibly making the search algorithm deliver better solutions at earlier stages. Unfortunately, the $O(n^{l+r+1})$ cost of PTA$(l, r)$ can well make it prohibitively high for microarray data problems in wrapper mode.

A clear avenue for further research is the setting of the free parameter, $\lambda$. It is our conjecture that an adaptive value may deliver better results. In this sense, the influence of past evaluations may be different at early or last stages of a search process.

CHAPTER

# 8 Combining Instance and Feature Weighting

In this chapter we present a novel method that aims at providing a more stable selection of feature subsets when variations in the training process occur. This is accomplished by executing an instance weighting (IW) process that assigns different importances to instances according to their outlying behavior; this weighting is a preprocessing step to the feature weighting (FW) that is independent of the learner or the specific FSS or FW algorithm. We report performance in two series of experiments: first using well-known benchmarking datasets and then some challenging microarray gene expression problems. Our results show increases in FSS stability for most subset sizes and most problems, without compromising prediction accuracy.

## 8.1 Introduction

Let's recall the definition of the hypothesis margin seen in Chapter 2:

$$\theta_S(\mathbf{x}) = \frac{1}{2}\left(|\mathbf{x} - \mathbf{nearmiss}(\mathbf{x})| - |\mathbf{x} - \mathbf{nearhit}(\mathbf{x})|\right) \tag{2.53}$$

With this definition we can see RELIEF as a filter algorithm that uses the hypothesis-margin concept in Eq. 2.53 to assess the importance of each feature in a dataset $D$ as the accumulated influence that each feature has in computing the margin of every instance in $D$ [38]. RELIEVEDF[41], a deterministic variant of the algorithm, picks one instance at a time and computes the hypothesis margin of each feature independently, accumulating the feature-wise distances to its nearest hit and nearest miss. As a result, the weight $w_j$ given to feature $X_j$ is its average distance to the selected neighbors:

$$w_j = \sum_{n=1}^{N} \Big( |(\mathbf{x}_n)_j - m(\mathbf{x}_n)_j| - |(\mathbf{x}_n)_j - h(\mathbf{x}_n)_j| \Big), \qquad j \in \{1, \ldots, d\}. \tag{8.1}$$

As we have seen in section "Margin based feature selection (Simba)" of Chapter 2, SIMBA is a more recent feature weighing algorithm that assigns weights to features based on their contributions to the hypothesis margins of the instances [3]. Since better generalization is expected if instances have larger margins, one should favour features that *contribute* more to these margins.

## 8.2 Logistic instace weighting

Here we present a novel instance weighting mehtod as an alternative to the Margin Based Instance Weighting With the purpose of obtaining a more robust evaluation, the average margin between every instance in $D$ and all the rest can be calculated as seen in Eq. 2.53 introduced in Chapter 2 and reproduced here to improve legibility.

$$\theta_S(\mathbf{x}) = \frac{1}{2}\left(|\mathbf{x} - \mathbf{nearmiss}(\mathbf{x})| - |\mathbf{x} - \mathbf{nearhit}(\mathbf{x})|\right) \tag{2.53}$$

Instances $\mathbf{x}$ achieving highly positive $\theta_S(\mathbf{x})$ present good modeling behavior (being far from misses and close to hits), while those with highly negative $\theta_S(\mathbf{x})$ become outlying ones (surrounded by misses and far from hits). The presence or absence of these latter instances in a training sub-sample is therefore a source of unstability.

With our Logistic Instance Weighting function we aim to use the above properties to give more importance to instances that are far from the hypothesis margin expecting a higher level of stability of the resulting FSS. In order to obtain a bounded positive weight in $(0, 1)$, we use a logistic function:

$$\omega(\mathbf{x}) = \frac{1}{1 + \exp\left\{-\alpha\, z\left(\theta_S\left(\mathbf{x}\right)\right)\right\}}, \tag{8.2}$$

where $\alpha$ is a parameter controlling the slope, and $z(\cdot)$ is the *standard score* $z(x) = (x - \hat{\mu}_D)/\hat{\sigma}_D$, being $\hat{\mu}_D$ and $\hat{\sigma}_D$ the sample mean and standard deviation of $\theta_S(\mathbf{x})$, for all $\mathbf{x} \in D$, respectively. A suitable value for $\alpha$ will depend on the problem and the user's needs. As a default value, under the assumption that hypothesis margins loosely follow a Gaussian distribution, we propose to set $\alpha = 3.03$, which corresponds to assign a weight of 0.954 to an instance whose average margin is two standard deviations from the mean, that is $\theta_S(x) = 2\hat{\sigma}_D$.

In order to illustrate the procedure, a simple example is provided. Consider a 2D synthetic dataset containing $N = 30$ instances, obtained by equally sampling from one of two distributions: either $\mathbf{x} \sim \mathcal{N}(\mu_1, \Sigma)$ or $\mathbf{x} \sim \mathcal{N}(\mu_2, \Sigma)$, where $\mu_1 = (0, 0)^T$ , $\mu_2 = (0, 0.25)^T$ and $\Sigma = diag(0.01, 0.01)$. Fig. 8.1 shows the weighted dataset, which clearly assigns low values to instances close to the boundary between classes and those inside opposite-class region; and assigns higher values the farther from the boundary inside the proper-class region. This is consistent with the intuition that outlying instances are a source of instability and therefore, must be lowly rated.

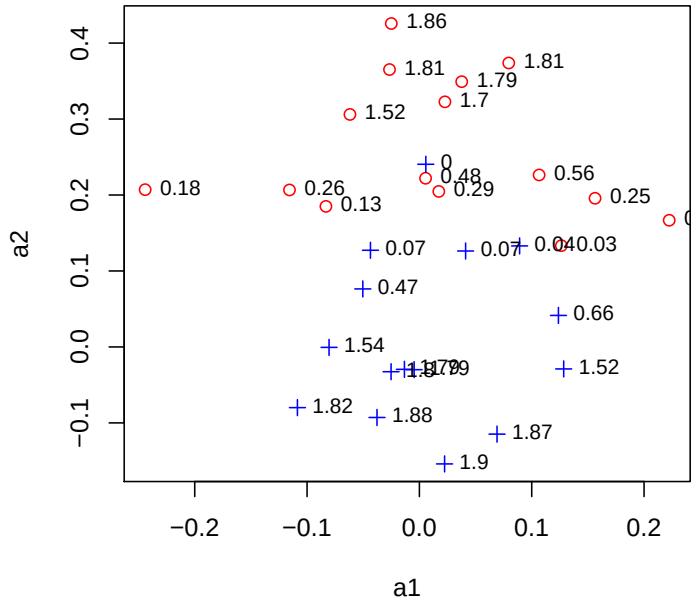



**Figure 8.1:** Importances assigned to the synthetic example data. These importances are computed using the formula in Eq. 8.2. The "+" and "o" symbols denote the two classes.

## 8.3 Combining Instance and Feature Weighting

One problem with the definition of the hypothesis-margin concept in Eq. 2.53 is the presence of outliers, or redundant or noisy features, which might mislead the margin calculus of an instance. The proposed method extends Simba to incorporate the instance weights, obtained both with the Margin Based Instance weighting by Han and Yu [29] introduced in Chapter 2 and with the Logistic Instance Weighting (LIW) presented in the previous Section, into the feature weights, to influence the way Simba behaves. We can consider different points of view for the combination of instance and feature weighting. First of all we can consider the way we execute the two algorithms:

1. The IW method is executed and the IWs are handed over to the FW method, which is in turn executed and the cycle recommences.
2. The IW method is executed as a subroutine of the FW method: when the FWs are updated (within the FW loop), so are the IWs.

In this chapter we have focused on the first approach leaving the second as future work as explained in Chapter 9.

Once we decided that way of classifying the interplay between both algorithms is by taking into account how do we make one influence the other. As we decided to run IW first we considered different approaches to modify the FW algorithm (i.e. (Simba) in our case) to take instance weights into account. The first way of doing so is by altering the way in wchich (Simba) selects the instances. We tested two different modifications (being Normal the unmodified version of the algorithm):

**Normal:** The unmodified version of the algorithm that uses all instances drawn in a random order.

**Sample:** Based selection on a probability distribution, according to the obtained weights (some instances may be selected more than once and some none).

**Order:** Iterate over every instance, and base the iteration order directly on the instance weights, from the instance with the largest weight downwards (all instances are selected exactly once).

Aiming at a deeper integration of IW and FW, we also propose to use the instance weights in the $\Delta$ calculation of the feature weights, which gives rise to another three variations:

**Normal$\Delta$:** Original SIMBA instance selection and use instance weights when computing the $\Delta$ feature weights.

**Sample$\Delta$:** Same as above but using Sample instance selection.

**Order$\Delta$:** Same as above but using Order instance selection.

In any case, we call the methods SIMBALIW: Simba with Logistic Instance Weighting and SIMBAMIW: Simba with Margin Based Instance Weighting (pseudo-code is shown in **Algorithm 8.1**).

## 8.4 Experimental Work

This section provides empirical evaluation of the proposed method. First, we illustrate SIMBALIW using a synthetically generated dataset; given that the truth behind the features is perfectly known, one has the possibility of assessing true performance. Then, the algorithm is tested to verify its real applicability, in three groups of problems: first using some well-known datasets from the UCI machine learning repository [2], then the ones used in a feature selection challenge organized by Guyon et al. during the *Neural Information Processing Systems 2003* conference (NIPS 2003) [76] and finally in widely-used cancer microarray data. These are different problems: first, *the number of features* is in the range of tens to a hundred for the former, and in the range of thousands for the latter; second, *the number of instances* is generally much lower for the microarray data.

The stability of an algorithm in selecting a subset of $k$ features out of the initial full feature size $d$ over a batch of $M$ runs can be evaluated using the Kuncheva [45] stability index $S_{Kuncheva}$ 2.34 that we described in Chapter 2.

**Algorithm 8.1:** SIMBALIW/SIMBAMIW $(D, \omega)$ (strategy can be either **sample**, **order**, **normal**)

$\mathbf{w} \leftarrow (1, 1, \ldots, 1)$ `// Feature weights`
**for** $n \leftarrow 1$ **to** $N$ **do**
  **if** *strategy is* ***order*** **then**
    let $\mathbf{x}$ be the instance ranked in position $n$ according to $\omega$
  **else if** *strategy is* ***sample*** **then**
    draw an instance $\mathbf{x}$ from $D$, according to the distribution $\omega / \|\omega\|_1$
  **else**
    let $\mathbf{x}$ be the $n$th instance of a random permutation of $D$
  **end**
  calculate $m(\mathbf{x})$ and $h(\mathbf{x})$ with respect to $D \setminus \{\mathbf{x}\}$ and the weight vector $\mathbf{w}$;
  **for** $i \leftarrow 1$ **to** $d$ **do**
    $\Delta_i \leftarrow \frac{1}{2}\left(\frac{(x_i - (m(\mathbf{x}))_i)^2}{\|\mathbf{x} - m(\mathbf{x})\|_{\mathbf{w}}} - \frac{(x_i - (h(\mathbf{x}))_i)^2}{\|\mathbf{x} - h(\mathbf{x})\|_{\mathbf{w}}}\right) w_i$
  **end**
  **if** *using* $\Delta$ *combination* **then**
    $\mathbf{w} \leftarrow \mathbf{w} + \omega(\mathbf{x})\boldsymbol{\Delta}$
  **else**
    $\mathbf{w} \leftarrow \mathbf{w} + \boldsymbol{\Delta}$
  **end**
**end**
$\mathbf{w} \leftarrow \mathbf{w}^2 / \left\|\mathbf{w}^2\right\|_\infty$ where $(\mathbf{w}^2)_i := (w_i)^2$

### 8.4.1 Synthetic Data

We first use a synthetic dataset designed to verify the performance of stable feature subset strategies [29] It consists of $M = 500$ training sets, each of the form $\mathbf{X}^m \in \mathbb{R}^{N \times d}$, with $N = 100$ instances and $d = 1,000$ features, for $m = 1, \ldots, M$. Every instance is equiprobably drawn from one of two distributions: $\mathbf{x} \sim \mathcal{N}(\mu_1, \Sigma)$ or $\mathbf{x} \sim \mathcal{N}(\mu_2, \Sigma)$, where

$$\mu_1 = (\underbrace{0.5, ..., 0.5}_{50}, \underbrace{0, ..., 0}_{950}), \quad \mu_2 = -\mu_1,$$

and

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \cdots & 0 \\ 0 & \Sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{100} \end{bmatrix},$$

being $\Sigma_i \in \mathbb{R}^{10 \times 10}$, with 1 in its diagonal elements and 0.8 elsewhere. Class labels are assigned according to the expression:

$$\mathbf{y}_i = \operatorname{sgn}\left(\sum_{j=1}^{d} \mathbf{X}_{i,j}\mathbf{r}_j\right), \quad \mathbf{r} = (\underbrace{0.02, ..., 0.02}_{50}, \underbrace{0, ..., 0}_{950}).$$

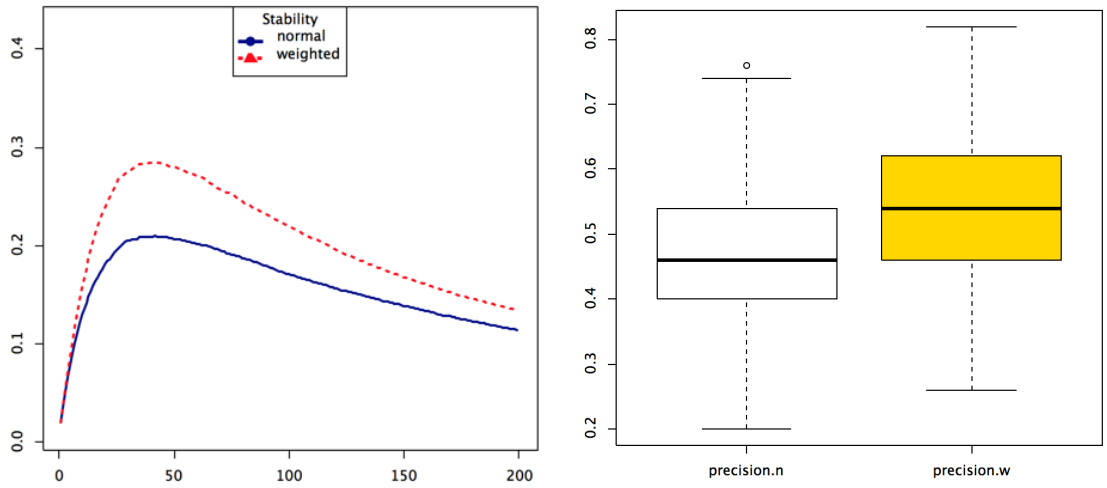



**Figure 8.2:** Feature stability on Han & Yu synthetic data. Left plot shows the average $S_{Kuncheva}$ over 500 repetitions of the process, as a function of increasing subset size (the bold line is the normal SIMBA, the dashed line is the weighted SIMBALIW version). Right plot shows the corresponding average precisions (n for normal, w for weighted).

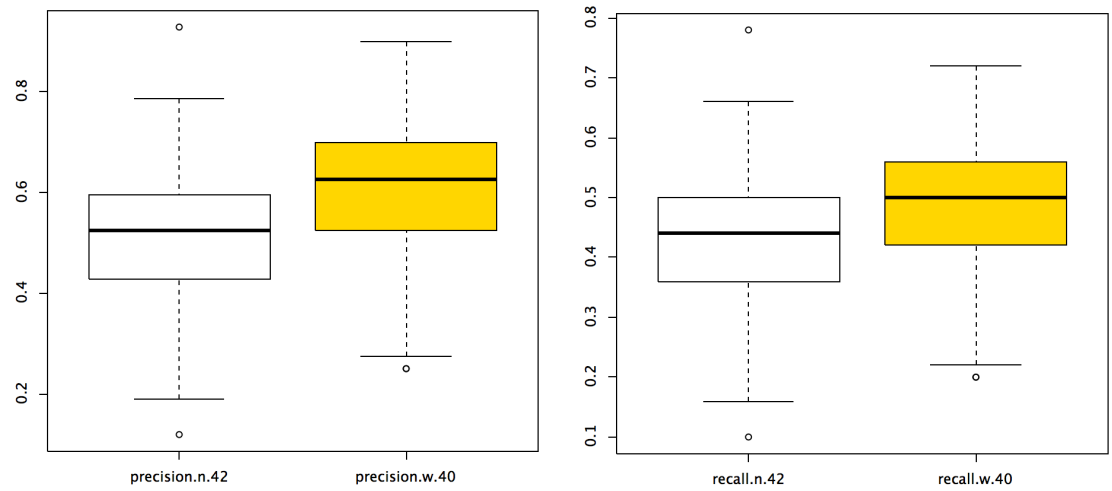



**Figure 8.3:** Boxplots of precision and recall at the point of maximum stability (42 features for SIMBA and 40 for SIMBALIW) on Han & Yu synthetic data.

The plots in Fig. 8.2 show the stability and accuracy results obtained by averaging 500 runs of independent artificial data generation, comparing SIMBA against SIMBALIW. The left plot is the average $S_{Kuncheva}$ as a function of subset size (size has been cut at 200 for clarity, the rest of the plot being similar to the shown slice). It can be seen the stability is increased at all subset sizes, topping in the 40-50 range (recall that this problem has exactly 50 relevant

features). The plot on the right shows boxplots of the distribution of *precision* over the 500 runs when the first 50 features are kept (thus we use the knowledge that only 50 features are relevant). Welch's t-test for a difference on the means (0.4722 vs. 0.5414) over the 500 independent results is highly significant ($p$-value $<$ 2.2e-16). Notice the *recall* is the same for both methods because there are exactly 50 relevant variables (out of the 1,000).

The plots in Fig. 8.3 do not use this knowledge. Rather, for both methods, we select the feature subset showing *maximum stability* and compute the corresponding average precision (left) and recall (right)[1].

We also reproduced the results for the xor problem presented in Section 2.4.2 as shown in **Figure 8.4**.We can see that all algorithms outperform RELIEF and that SIMBALIW seems to be the one that converges first to the correct feature weights.

[1] Recall = **True positives** / (**True Positives** + **False Negatives**); Precision = **True positives** / (**True Positives** + **False Positives**). A **True Positive** is a selected and relevant feature, a **False Negative** is a discarded and relevant feature, etc.

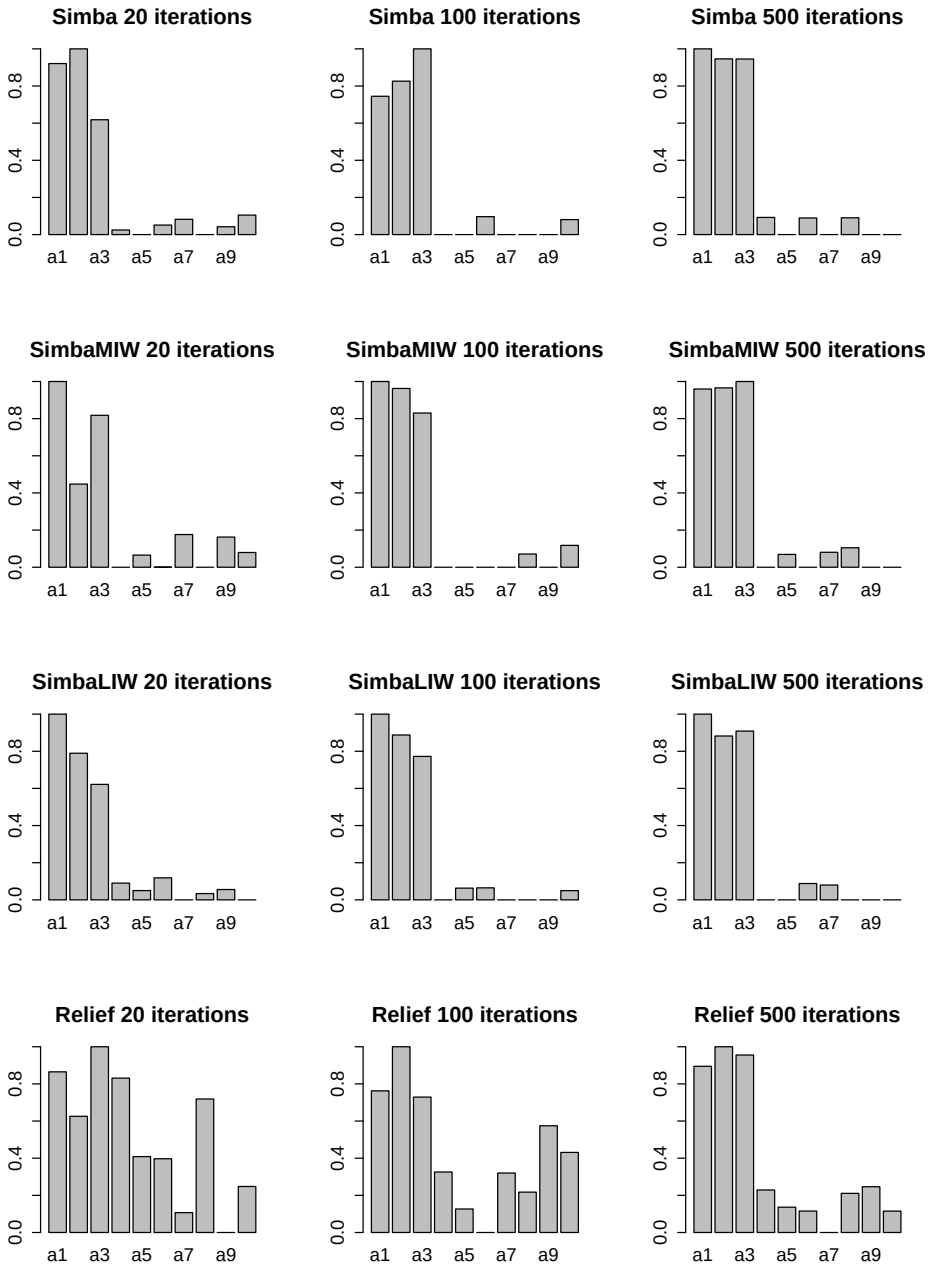



**Figure 8.4:** The weights Simba, SimbaLIW, SimbaMIW and Relief assign to the 10 features when applying on the xor problem.

### 8.4.2 Real Data

A collection of 15 UCI, 5 NIPS and 6 microarray datasets presenting a variety of diseases is used. The description of the data can be found in Section 3.1.

### 8.4.3 Experimental setting

The experimental setup consists of the two nested cross-validation loops described in Section 3.2. For every fold and repetition of the outer cross-validation loop, two feature-weighting processes are conducted with the same instances: one with the original SIMBA algorithm and one with our modified version taking instance weights into account. $S_{Kuncheva}$ is computed for every subset length at every partition loop and then averaged over the 10 times. Once the features have been obtained we test the obtained feature weights using a modified k-NN classifier as shown in **Algorithm 8.2** that accepts both instance and feature weights, recording prediction accuracy on the leftout test parts. We use these weights to perform an **inner** 5x2-fold cross-validation with the purpose of estimating the prediction error of each classifier. This error is then computed for each fold to compare the feature sets selected by both SIMBA, SIMBAMIW and SIMBALIW. This modified k-NN classifier uses the feature weights to ponderate the distance calculation between two instances as shown in Eq. 8.3 (see line 3 in the algorithm). In addition instead of using a majority voting as the original k-NN does to compute the label of the test instance it uses the instance weights to give make more relevant instances more influent in the voting (see line 8 in the algorithm). By using an algorithm that accepts feature weights we overcome the need of finding a suitable feature set given the resulting weights of the FW process as we did in our previous paper [56]. If we wanted to use the traditional version of k-NN at this point we would have to decide a size $k$ of the selected feature set, order the features according to their weights and keep the first $k$ or execute the classifier for every possible size and keep the feature set size that gave the better prediction accuracy.

**Algorithm 8.2:** Instance and Feature Weighted k-Nearest Neighbor Algorithm

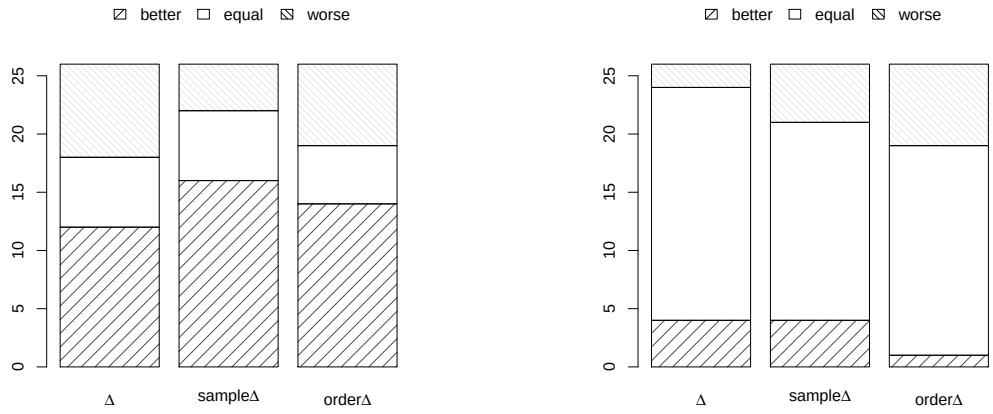
**Input** : Training set $T = \{\mathbf{x}_1, ...\mathbf{x}_N\}$, constant $k$, instance weights $\omega$, feature weights $\mathbf{w}$, new instance $\mathbf{x}_t$ to be classified
**Output**: Class prediction for $\mathbf{x}_t$

1 Initialize all $c_i \in c$ to 0;
; // class counters
2 **foreach** $x_n \in T$ **do**
3 $dist_n \leftarrow d_{\mathbf{w}}(\mathbf{x}_n, \mathbf{x}_t)$ ; // where $d_w$ is the weighted distance in Eq. 8.3
4 **end**
5 Sort $dist$ in descending order;
6 $I_k \leftarrow$ nearest $k$ instances according to $dist$;
7 **foreach** $\mathbf{x} \in I_k$ **do**
8 $c_i \leftarrow c_i + \omega(\mathbf{x})$ ; // where $i$ is the class of $\mathbf{x}$ and $\omega(\mathbf{x})$ its instance weight
9 **end**
10 **return** $\arg\max_{i \in c} c_i$;

Here $d_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ is the weighted distance between instances $\mathbf{x}$ and $\mathbf{y}$ using the feature weights $\mathbf{x}$ as seen in Eq. 8.3.

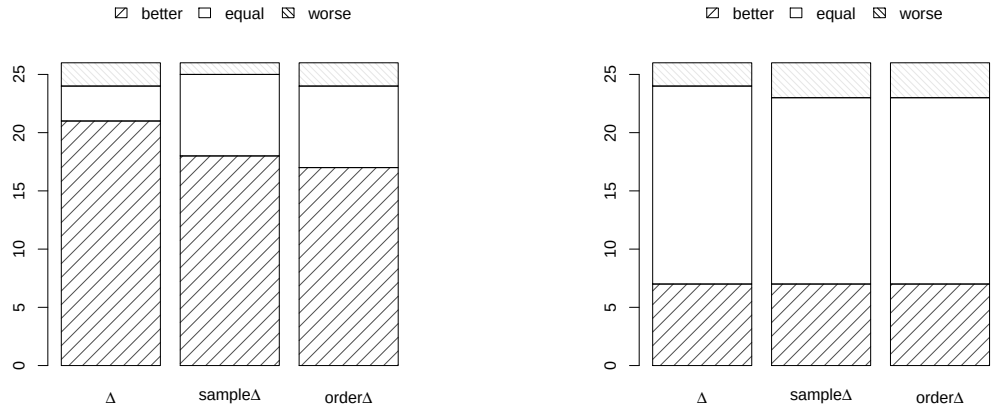
$$d_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d} w_i (x_i - y_i)^2} \tag{8.3}$$

We have moved the detailed results for these experiments to Appendix C for the sake of simplicity. There we can find Figures C.1 to C.9 that show results on feature subset selection stability on UCI, NIPS and microarray data for normal, sample and order delta weight combinations. They also display average test errors when training a classifier using the resulting feature weights. To help summarizing the results we also present Tables C.1 to C.36 with the average and starndard deviation of the classification errors and the average and standard deviation of the stability for each algorithm and weight combination. We have summarized the results shown in those tables to Figures 8.5 and 8.6 where we can see the number of problems (including UCI, NIPS and microarray) which the modified versions of the FSS algorithm had better/equal/worse stability results and also the number of problems which the classification error of the resulting feature sets was better/equal/worse. We can clearly see that both modifications lead to more stable results most of the time. In fact, SimbaMIW is only significantly less stable than Simba in the NIPS Challenge Madelon dataset when using sample$\delta$ weight combination. We can also see that SimbaMIW delivers the most stable feature sets of the three, followed by SimbaLIW. Being the latter the algorithm leading to lower classification errors on average.




**(a)** Stability **(b)** Classification error

**Figure 8.5:** Number of problems where SimbaLiw was better/equal/worse than non-modified Simba regarding stability and classification error.




**(a)** Stability **(b)** Classification error

**Figure 8.6:** Number of problems where SimbaMiw was better/equal/worse than non-modified Simba regarding stability and classification error.

Excluding small variations at the very first and last iterations, the same general trend can be observed in all sets of plots, showing that stability is enhanced for most subset sizes and virtually all problems except for the UCI datasets when using the order delta combination. It is interesting to note that the small or null gains correspond to very small (sizes $1-5$) or large (near full size) subsets, where the set of possible combinations is much smaller. Remarkably, on all other sizes, both SimbaMIW and SimbaLIW choose feature subsets in a more stable way when the training set changes.

It is beyond the intention of this study to search for the best absolute subset (or best subset size) for each problem. Rather, the motivation is to study stability for all sizes and a glimpse at possible differences in test accuracy. In this sense, as it could reasonably be expected, performance varies for the different chosen sizes. Some problems seem to benefit from an aggressive FSS process and others the other way around.

## 8.5 Generalizing to other feature weighting algorithms

These IW algorithms have been chosen specifically to help Simba as they use the same margin concept but here we want to apply the same concept of IW and FW combination to different FW algorithms.

Here we present a framework that can be used to combine any IW and FW algorithms. We propose to use the instance weights to resample the input data that the FW algorithm receives. We will use a sample with replacement method similar to the *sample* and *order* stratgies described above for Simba as a previous step to calling the FW algorithm. The steps can bee seen in **Algorithm 8.3**. Of course, the *order* version only makes sense for algorithms

that iterate over the instances such as RELIEF or RANDOMFORESTS [7]. The versions we are using of these algorithms iterate over the selected instances in the same order that they appeared in the original dataset. Our version of RELIEF takes a number of random instances over the original dataset but then iterates over them in the original order. Same holds true for the RANDOMFORESTS algorithm. This algorithm selects the instances to build the trees with the bagging [6] algorithm, thus randomly. But once a bag of instances is selected it iterates over each instance in the bag in the same order they appeared in the original dataset. As an example, if the original dataset contained the instances $\{x_1, x_2, x_3, x_4, x_5\}$ a possible bag of instances could be $\{x_4, x_2, x_3, x_2, x_5\}$. In this example RANDOMFORESTS would iterate over the instances in this bag as $\{x_2, x_2, x_3, x_4, x_5\}$ preserving the original order among them. Other FW algorithms based on the mutual information or the joint entropy between a feature and the class that do not iterate over instances can also use a resampled input data set with the *sample* strategy. The frequencies of the features' values will vary as the input data may contain repetitions of certain instances and may omit others.

**Algorithm 8.3:** IW and FW combination framework

**Input** : Training set $T = \{\mathbf{x}_1, ...\mathbf{x}_N\}$, classes $C$ (where $\mathbf{x} = \{x_1, \ldots, x_d\}$ is a particular instance with $d$ features)
**Output**: Feature weights: $\mathbf{w} = \{w_1, \ldots, w_d\}$

```
1 Instance weights: ω = iw(T, C) ;          // where iw(·) can be any IW algorithm
2 for n ← 1 to N do
3     if strategy is order then
4         S'[n] ← the instance ranked in position n according to ω
5     else if strategy is sample then
6         S'[n] ← an instance x from D, drawn according to the distribution ω/‖ω‖_1
7 end
8 w = fw(S', ω) ;                           // where fw(·) can be any FW algorithm
```

We have tested this framework with three well known FW methods in addition to the experiments with SIMBA presented above: RELIEF, RANDOMFORESTS, IG (information gain) and 1R [31]. Figures 8.7 to 8.12 show the summary results for the different algorithms and strategies. Appendix C contain all the detailed results for each algorithm in Figures C.28 to C.72 and Tables C.37 to C.84.

By looking at the charts we can see that for the majority of the problems the stability is improved while keeping or improving prediction power.

The exceptions are RELIEF and RANDOMFORESTS with the *sample* strategy. By looking at the detailed results the bad results are most of the time concentrated in the problems with lots of features and few examples such as the microarray ones. A possible explanation is that we are losing some instances in the resampling. As we stated before the *sample* strategy draws random examples with replacement from the original dataset with a probability proportional to the instance weights. We also pointed out that this may lead to omitting some of the instances. In these kind of problems where only a few instances are present omitting one may have a big impact both on FW stability and on classification error. A possible way of overcoming this situation would be to make a deeper integration of both algorithms as we did


**(a)** Stability **(b)** Classification error

**Figure 8.7:** Number of problems where dataset resampling according to rliw instance weights delivered better/equal/worse results regarding stability and classification error for RELIEF.

with SIMBA to make the internal resampling aware of the instance weights. We will leave this as future work.

We also note that IG is already almost 100% stable so applying IW makes no sense in most cases. But even with this situation IW manages to improve stability. IW only affects IG in the choice of the most relevant features as seen in the detailed experiments Figures C.66 and C.67. In some cases the original version presents instability in the ranking of these most relevant features, all of these situations are solved by IW. But unfortunately there are other situations where the original algorithm was completely stable and by using IW some instability is introduced precisely in the order of these high relevant features.

The best results are achieved with the *order* strategy for RELIEF and RANDOMFORESTS. The former significantly outperforms the original algorithm in 55% of the problems and underperforms it in 21%. Even better results are achieved with the latter, outperforming the original version in 63% of the problems and underperforming it in 15%. Regarding the IW algorithm, RLIW is clearly better for RELIEF while MBIW helps RANDOMFORESTS more. This again might be because we designed RLIW to use the exact margin definition as RELIEF does. This also suggests us that by further studying the RANDOMFORESTS algorithm a better IW technique could be found in addition to deepening the integration of the instance weights.

## 8.6 Conclusions

In this chapter we have introduced a new method for improving the stability of feature subset selection algorithms, which draws upon previous algorithmic work on feature weighting and hypothesis margins for instances. Our strategy uses a double set of weights, one for the


**(a)** Stability **(b)** Classification error

**Figure 8.8:** Number of problems where dataset resampling according to mbiw instance weights delivered better/equal/worse results regarding stability and classification error for RELIEF.

features and another one for the instances. The suitability for standard feature selection practice has been assessed using data from different environments: microarray gene expression and real-world datasets from the UCI repository and from the NIPS 2003 feature selection challenge. We have presented a novel modification of SIMBA FW algorithm that takes instance weights into account that clearly outperform the original algorithm on most cases. Again we will cite the *No free lunch* theorems that state that if a machine learning algorithm achieves superior results on some problems, it must pay with inferiority on other problems. We also defined a framework to be able to apply the idea of IW and FW combination for any two choices of algorithms and tested it with a number of classic algorithms. We have proven that the least stable ones (i.e. RELIEF and RANDOMFORESTS) can be substantially more stable with this technique and even the more stable IG and 1R algorithms can be slightly improved. It is also important to note that the prediction error of a classifier trained with the resulting feature weights is not negatively affected and even improved in some cases. It could also be argued that models showing high stability are inherently more robust to input variance, and therefore more advisable to rely upon, despite showing lower prediction accuracy in certain cases. The reported results on the non-modified algorithms used in this combination framework suggest that there is still more room for stability enhancement by a deeper integration of the two algorithms.

The present work offers a number of interesting avenues for further research. First, there are several alternative ways to combine the instance weighting idea and the SIMBA feature weighting algorithm. In particular, the instance weights can be updated at each iteration, given that the feature weights are re-computed, which would lead to a synergetic process. Second we devised an opportunity for even more stability improvement by modifying the FW


**(a)** Stability **(b)** Classification error

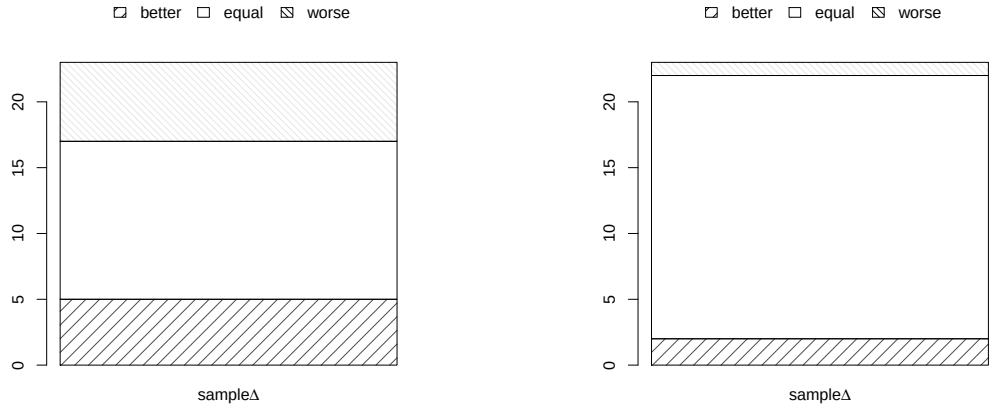
**Figure 8.9:** Number of problems where dataset resampling according to rliw instance weights delivered better/equal/worse results regarding stability and classification error for RandomForests.

algorithms so modifications for Relief and RandomForests can be explored. Finally, we have seen that by using an IW schemes that uses the same information as the FW algorithm better results are achieved. This being the case of RLIW and Relief, so other IW schemes could be designed to improve RandomForests.

Part of this work has been submitted to ESANN 2016 and is under review as listed in Appendix D.

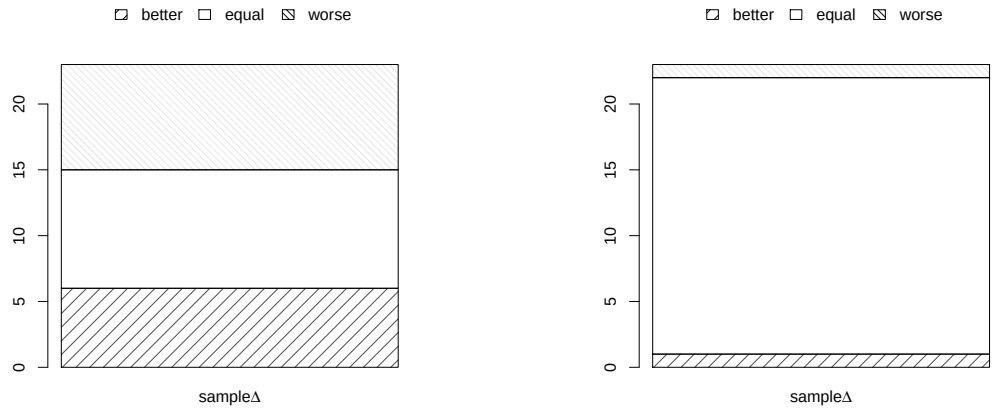

**(a)** Stability **(b)** Classification error

**Figure 8.10:** Number of problems where dataset resampling according to mbiw instance weights delivered better/equal/worse results regarding stability and classification error for RandomForests.



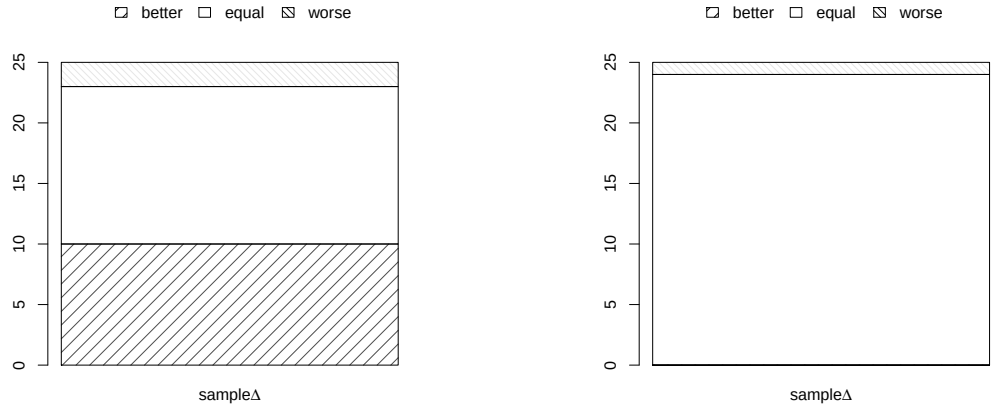

**(a)** Stability **(b)** Classification error

**Figure 8.11:** Number of problems where dataset resampling according to rliw instance weights delivered better/equal/worse results regarding stability and classification error for InformationGain.



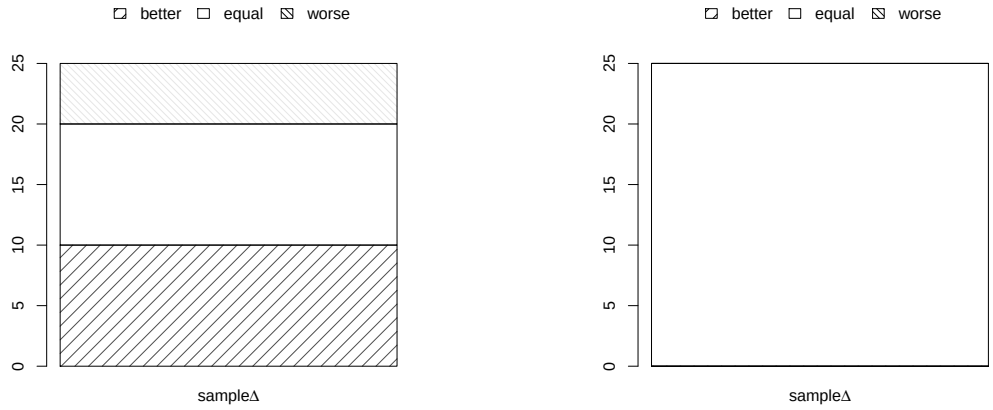

**(a)** Stability **(b)** Classification error

**Figure 8.12:** Number of problems where dataset resampling according to mbiw instance weights delivered better/equal/worse results regarding stability and classification error for InformationGain.




**(a)** Stability **(b)** Classification error

**Figure 8.13:** Number of problems where dataset resampling according to rliw instance weights delivered better/equal/worse results regarding stability and classification error for 1R.




**(a)** Stability **(b)** Classification error

**Figure 8.14:** Number of problems where dataset resampling according to mbiw instance weights delivered better/equal/worse results regarding stability and classification error for 1R.

Chapter

# 9 Conclusions and future work

In this thesis we have widely studied feature subset selection (FSS) and feature weighting (FW) stability. Stability for these algorithms is a paramount subject both to have a better description of the input data and to allow better comparisons of the different methods. This is specially important for the DNA microarray problems that are used throughout the thesis. These problems are characterized by having a large number (i.e. tens of thousands) of features and a small (i.e. hundreds) of examples. Having unstable FSS results does not help doctors trying to study which genes are the ones that should be analyzed to help them have an early diagnosis. We have found out that stability is not a topic that had been deeply studied. In fact we have identified several weaknesses to some of the previously proposed stability measures. Once a stability measure have been chosen, we have outlined and studied different sources of instability for FSS algorithms: The effect of redundant features, the intrinsic instability introduced by the usage of random variables to assess feature importance, the effect of randomly choosing the instances at each iteration, the order in which features are evaluated in sequential algorithms or the different contributions two instances may have to stability (e.g. a patient living near a source of radiation may introduce a great amount of instability when we want to assess the prediction power of each gene and we don't have a lot of patients). Different experiments have been conducted to test hypothesis on each of the instability sources with a robust experimental framework consisting of two nested loops of cross-validation. The outer loop gives us 10 runs of the FSS or FW algorithms and a set of instances that we use to assess the prediction power of the selected features that have not been used in the FSS/FW process. Inside the FSS/FW process another loop of cross-validation is performed to train the inducer using a set of instances and assess the importance of the features with another set. We used statistical significance tests to compare the results of two FSS/FW algorithms. Our experiments on each of the hypotheses used this experimental setup and a set of 25 real life problems obtained from the UCI machine learning datasets, the NIPS 2003 feature selection challenge datasets and some public DNA microarray problems. In addition for each of the modifications we have proposed various variants have been tested to make our results even more robust. In the case of *wrapper* FSS we have tested our proposals using three different

inducers (1NN, LDA and SVM) and in the case of *filters* we have tested our modifications with five different FW algorithms (RELIEF, IG,RANDOMFORESTS, 1R and much deeply with SIMBA). Below we describe the principal conclusions:

- Fist of all we have had a look on how to define stability and analyzed different measures to do so. As we have seen in Chapter 2 there seems to be a clear winner regarding the assessment of FSS stability. $S_{Kuncheva}$ presented by Kuncheva [45] in 2007 remains unbeaten as it has correction by chance and it is the more sensible of the presented stability measures. On the other hand if we want to measure the stability of FW methods we have two alternatives. Either we set a cutting point for the number of selected features and assess the stability of the resulting FSS problem or we use a rank order correlation index as we did in 8 such as the Spearman's rho, Kendall Rank or Gini Index. We have shown that many of the alternative stability measures such as $S_{Kalousis}$ or $S_{Dunne}$ are not a good choice as they favor feature subsets that are either nearly empty or nearly complete.

- We have also presented a robust framework in Chapter 3 that allows us to assess different FSS and FW algorithms and evaluate both their stability and the prediction power of the resulting feature sets (or feature weights) by using a double loop of 5x2cv. This framework allows to evaluate different algorithms even when the size of the training data set is very small making it very suitable for problems, such as the gene expression microarray problems used in this thesis, with very few instances and a lot of features.

  By using this framework and with the stability measure in place we have explored several ways to improve stability and performance of FSS and FW both *wrapper* and *filter* algorithms.

- Regarding *wrappers*, we presented two novel modifications of the wrapping algorithm: The remainder set aware (RSA) and the accumulated evidence ($SBG^+$) in Chapters 6 and 7. For each modification we have tested various inducers (1NN, LDA and SVM) to asses the quality of the proposed modification. We have proven that both modifications render improved stability results for most of the problems and also better performance regarding classification error of the resulting inducer.

- We introduced various novel combinations of instance weighting and feature weighting for *filter* algorithms in Chapter 8. We have tested the existing margin based instance weighting (MBIW) strategy and proposed the novel logistic instance weighting (LIW) one targeted to improve the SIMBA algorithm —an improved version of the well known RELIEF algorithm. We also modified SIMBA to use the instance weights in two different ways: the first one by influencing the way the algorithm selects the instances at each iteration and then a deeper integration that weigh the instance contributions in the FW calculations. These combinations of SIMBA with MBIW and LIW —which we have called SIMBALIW and SIMBAMIW— lead to more stable results for the FSS and FW problems without incrementing the computational cost in a significant way as some of the other stability improvement methods do such as ensemble FS. We have also proved that this improvement on stability results in feature sets with equal and even higher prediction power when used by a classifier. We have also defined a framework to combine any IW and FW algorithms by resampling the input data of the FW algorithm according

to the results of IW using two different strategies: sorting the input data according to the ranking of each instance (*order*) and sampling instances at random with replacement with the probability for each instance to be chosen at each step proportional to the weight assigned by the IW algorithm (*sample*). We have tested this framework with four classic FW algorithms: ReliefF, RandomForests, InformationGain and 1R. We have proven that stability is very significantly increased for the two former ones. In the case of the two latter we have seen that, as they are performing very simple computations, their stability is already near 100% so it can seldom be improved. Even in this situation the combination with IW managed to improve stability in some cases. We also have proven that by training an inducer with the set of resulting features can lead to significantly higher performance, especially for ReliefF and the *order* strategy where almost 30% of the problems achieved significantly lower classification error percentages.

The topic of feature selection stability is a large one, and there are many avenues for new work towards achieving stability –both of selected features and model predictions– without sacrificing overall accuracy through feature weighting and observation weighting.

Current work concentrates in developing further some of the methods already presented in the thesis. In all cases, the general idea is to add more flexibility by generalizing the methods and at the same time to carry further the possibilities of the methods.

We next detail four current developments; these have the advantage that can be developed independently of one another. We intend to include in the final document those that lead to increased performance.

Throughout this chapter, we use the abbreviations IW (Importance weighting, for observations) and FW (Feature weighting).

## 9.1 Use a generalized distance

The standard unweighted Euclidean distance is not the only possible choice to measure distance between two observations drawn from two probability densities $f_1$ and $f_2$, corresponding to different groups or classes in a classification setting.

We consider a general distance introduced in [11]:

$$d_\alpha(f_1, f_2) = -\log \int f_1(\mathbf{x})^\alpha f_2(\mathbf{x})^{1-\alpha}\, d\mathbf{x},\ \alpha \in [0,1], \mathbf{x} \in \mathbb{R}^d$$

When $\alpha = 0.5$, it reduces to $-\log \int \sqrt{f_1(\mathbf{x}) f_2(\mathbf{x})}\, d\mathbf{x}$, sometimes referred to as the Bhattacharyya measure of affinity. If $f_1$ and $f_2$ are multivariate Gaussian densities with means $\mu_1, \mu_2$ and covariance matrices $\Sigma_1, \Sigma_2$ (resp.), then:

$$d_{0.5}(f_1, f_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \left( \frac{|\Sigma|}{\sqrt{|\Sigma_1||\Sigma_2|}} \right)$$

where $\Sigma = \frac{1}{2}(\Sigma_1 + \Sigma_2)$ –see [35]. This distance easyly generalizes to accommodate more than two classes. The idea is to use $\Sigma_i, \Sigma_j$ when computing the distance between two observations from classes $i$ and $j$. This measure takes into account the covariance structure of the classes and as such it is a much more informed one than standard Euclidean distance. In practice, $\Sigma_i$ and $\Sigma_j$ are replaced by the sample covariance matrices $\hat{\Sigma_i}$ and $\hat{\Sigma_j}$.

The aim is to use this generalized distance $d_{0.5}$ in place of standard Euclidean distance, both in the definition of the hypothesis margin and in the SIMBA algorithm. This step is not free of issues, which are specified and dealt with below.

We are currently extending the hypothesis margin of an instance $\mathbf{x}$ to a set of data points $D$ to:

$$\theta_D(\mathbf{x}) = \frac{1}{2}\Big(d_{0.5}(\mathbf{x}, m(\mathbf{x})) - d_{0.5}(\mathbf{x}, h(\mathbf{x}))\Big)$$

where $m(\mathbf{x})$ and $h(\mathbf{x})$ are the **near hit** and **near miss**: the instances in $D$ nearest to $\mathbf{x}$ with the same and with a different class label, respectively. Specifically, the first $d$ uses $\Sigma_i$ and $\Sigma_j$, being $i$ the class of $\mathbf{x}$ and $j$ that of $m(\mathbf{x})$. Since –by definition– the classes of $\mathbf{x}$ and $h(\mathbf{x})$ coincide, the same (common) covariance matrix $\Sigma_i$ is used in the second $d$ (being $i$ the class of $\mathbf{x}$). The same method is followed for the co mputation of $m(\mathbf{x})$ and $h(\mathbf{x})$ themselves.

There is a first concern in what regards computational requirements. The new distance demands a matrix inversion (actually one for every posible class pair); it also requires matrix multiplication operators every time the distance is evaluated. The proposed solution is to perform a coordinate transform for every $X \sim N(\mu, \Sigma)$ to $Y \sim N(0, I)$. Departing from the Schur decomposition $\Sigma = Q\Delta Q^T$, where $Q$ is an orthogonal matrix (whose columns are the eigenvectors of $\Sigma$) and $\Delta = diag(\lambda_1, \ldots, \lambda_d)$ contains the eigenvalues of $\Sigma$.

Now

$$(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) = (\mathbf{x} - \mu)^T (Q\Delta Q^T)(\mathbf{x} - \mu)$$

$$= (\mathbf{x} - \mu)^T Q^T \Delta^{-1} Q(\mathbf{x} - \mu) = (QA(\mathbf{x} - \mu))^T I(QA(\mathbf{x} - \mu)) = y^T I y$$

where $A = diag((\lambda_1)^{-1/2}, \ldots, (\lambda_d)^{-1/2})$. Therefore we propose to preprocess the data $\mathbf{x}_n$ from a certain class $j$ as $\mathbf{y}_n = Q_j A_j \mathbf{x}_n$, being $Q_j A_j$ the matrices derived from $\Sigma_j$. Note that the transformation $\mathbf{y}_n = Q_j A_j(\mathbf{x}_n - \mu_j)$ is sometimes known as the *whitening transform*. In our case there is no need to center the data, because this could lead to a loss in separability (although both possibilities will be explored). The new determinants now all evaluate to 1.

We can now use the FW methods to the transformed data. In particular, both RELIEF and SIMBA assume that the covariance structure in the groups is precisely the identity matrix (this is implicit in the use of unweighted Euclidean distance).

A second concern was found in the way standard SIMBA uses the FWs. These are used to weight the previously unweighted Euclidean distance. Since the proposed distance is a fully weighted one, both sets of weights "interfered". These are different kinds of weights: those

in the covariance matrices reflect the statistical distribution of the classes, while the FWs reflect the importance of every feature to separate the classes. Now that SIMBA gets truly uncorrelated data, this issue is settled.

## 9.2 Use of the learned weights

The primary aim of computing IWs and FWs has been to enhance the stability of the selected features. Feature selection can often be considered part of model construction and becomes an important step, but not the only one. We propose to use the sets of learned weights also for building the models, by incorporating them (as far as possible) into the learning algorithm itself.

Note that, in so doing, the whole process still falls into the filter approach (as opposed to the wrapper one), inasmuch the classifier's predictions are not fed back to guide the process.

Another issue is found in the study of the stability of the predictions (regardless of accuracy), across the data partitions. In two class problems, we can calculate the correlation between two classifier's outputs as:

$$\rho = \frac{N_{11}N_{00} - N_{01}N_{10}}{\sqrt{(N_{11}+N_{10})(N_{01}+N_{00})(N_{11}+N_{01})(N_{10}+N_{00})}},$$

where $N_{ab}$ is the number of predicted observations for which one classifier gives a value $a$ and the other gives a value $b$, and $a, b \in \{0, 1\}$. This measure should be extended to more than two classes. A trivial way would be to consider an agreement when predictions coincide (a '1') and a disagreement when predictions do not coincide (a '0') and then apply the previous measure.

The incorporation of the IWs seems the most general avenue. Many learning methods are willing to accept (or to be extended with) information on how important every observation is to the fit. These include classical regression methods (like Logistic regression or PLS [85]), modern regression methods (like the LASSO or the Elastic net), nearest neighbours, discriminant analysis or even the Support Vector Machine [86]. The references given correspond to specific proposals in the literature with this aim.

## 9.3 Explore the interplay

The study of possible synergies between the IW and the FW methods opens a wealth of possibilities. From a very general point of view, the IW method uses the FW method and vice versa. Since the former does not imply an iterative procedure, two basic schemes can be derived:

1. The IW method is executed and the IWs are handed over to the FW method, which is in turn executed and the cycle recommences.

2. The IW method is executed as a subroutine of the FW method: when the FWs are updated (within the FW loop), so are the IWs.

The way the IWs are used to influence the FW method has been a specific subject of this thesis. We have introduced the 'order' and 'sample' methods into the SIMBA algorithm with good results. We are now studying a third method, described below.

In a sense, the 'order' and 'sample' methods are extremers in that the former is too deterministic and the latter too stochastic. We pretend to derive a midway method by first sampling a number $q$ of observations uniformly at random, and then choosing the one with the largest IW. An inspiration for this procedure is found in evolutionary algorithms, where a form of selection called *tournament selection* is used to select individuals for reproduction [51]. As a first idea, we propose to use $q = \sqrt{N}$, being $N$ the total number of observations.

## 9.4 Optimization of SIMBA

SIMBA's original FW update rule is derived as an optimization problem. However, it is not entirely solved as such. First, the information supplied by the gradient $\bigtriangledown e(\mathbf{w})$ of $e(\mathbf{w})$ is typically used in standard optimization methods like gradient descent (GD), which are iterated until convergence. Therefore a full GD should be performed at every SIMBA iteration step. Moreover, SIMBA uses a constant learning rate of 1, which could be largely suboptimal. Second, GD is a first-order method, having little access to the local curvature of the function being optimized.

This proposed line of research entails the proper analysis of $e(\mathbf{w})$ as a function of the utility function $u$ (e.g., if $u$ is continuously differentiable, then so is $e(\mathbf{w})$). If the final form of $e(\mathbf{w})$ is quadratic on $\mathbf{w}$, then algorithms like Newton-Raphson (NR) [37] will find the global optimum in one step. If, albeit not quadratic, $e(\mathbf{w})$ is convex, then NR can still be applied; it will take a small number of iterations to converge, but the existence of a unique optimum is still guaranteed. In a nutshell, to maximize $e(\mathbf{w})$ we need to iterate the step:

$$\mathbf{w}^{(\text{new})} \leftarrow \mathbf{w}^{(\text{old})} - H_{\mathbf{w}}^{-1} \bigtriangledown e(\mathbf{w})$$

where $\bigtriangledown e(\mathbf{w})$ is the gradient vector and $H_{\mathbf{w}} = \bigtriangledown(\bigtriangledown e(\mathbf{w}))^T$ is the Hessian matrix, both evaluated at $\mathbf{w}^{(\text{old})}$. This algorithm is the standard in many classical methods, like Logistic regression.

## 9.5 Progressively weighted Simba

In Chapter 5 we proposed a modification of RELIEF (DRELIEF) that was very similar to SIMBA yet proved to be inferior to it. But then we introduced the concept of progressively increasing the effect of the previously computed feature weights on the computation of the weights in the current iteration that significantly improved the previous version.

It would be interesting to apply the progressive approach to the influence the previously computed weights has to SIMBA which may lead to an algorithm which outperforms both its ancestors.

## 9.6 Study the effect of redundancy and importance on FSS stability

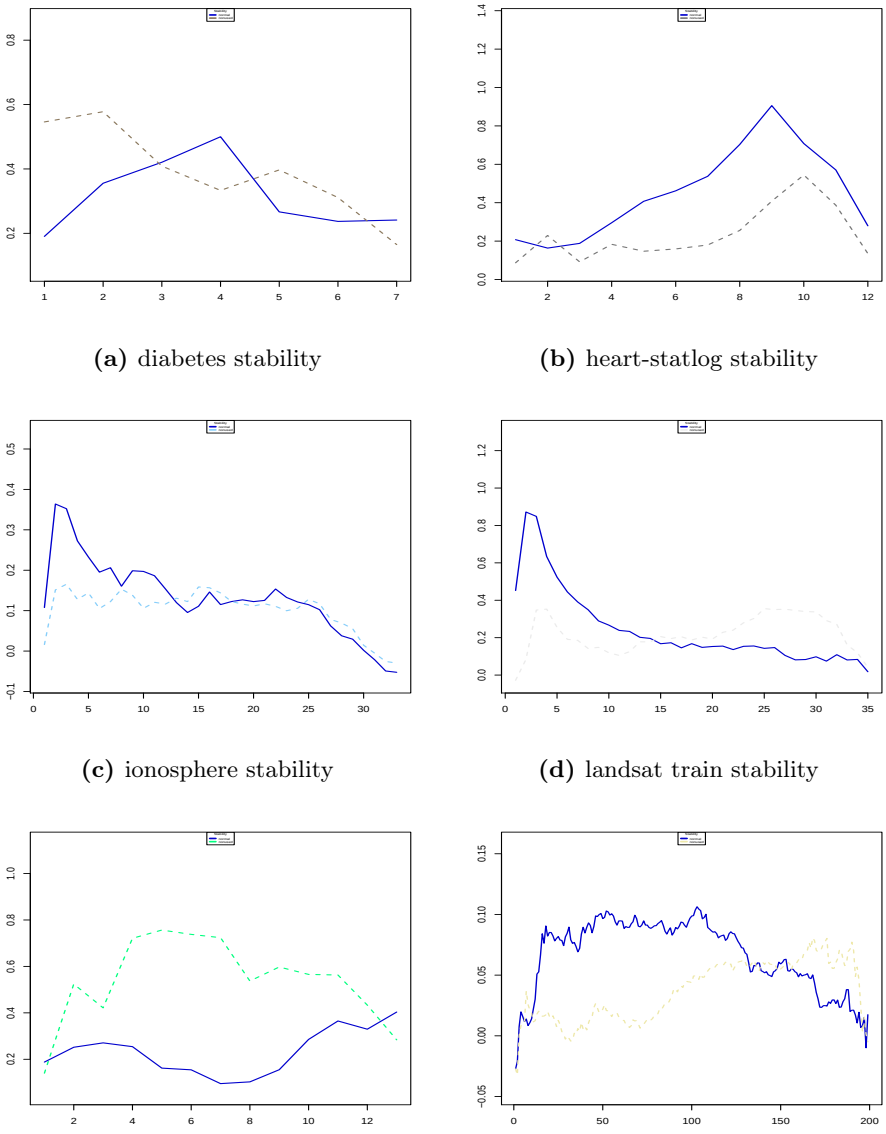
In Chapter 4 we presented a theoretical definition of a redundancy index based on Markov Blankets. Several algorithms have been developed that try to approximate the Markov Blanket definition that have a reasonable cost. Schlüter [67] wrote an exhaustive survey of the different algorithms comparing their properties. One interesting research line would be to study the correlation of feature redundancy with the instability of FSS and FW algorithms, aimed at proposing modifications of these algorithms that cope with redundancy, thus leading to more stable results. Concerning the study on feature importance, despite the fact that as yet it is impractical –given that it requires the Bayes error–, a good starting point would be to use artificial problems where this knowledge is available and test the definition and its influence on instability of FSS and FW.

## 9.7 Analysis of the influence of the inducer in remainder subset aware FSS

We described a way to improve stability of *wrapper* sequential FSS algorithms in Chapter 6. We explored the effect of taking the remainder subset of features into account when selecting the next included (or discarded) feature at each step of Sequential Forward (or Backward) Generation algorithms. This modification often lead to improvements of stability but have substantially different results when different inducers are used. Another line of future work could be to test when this modification is most helpful and to provide a theoretical explanation of the reasons why it is or it is not helpful for a particular algorithm and inducer.

Appendix

# A Remainder Subset Awareness detailed results



**(a)** diabetes stability

**(b)** heart-statlog stability

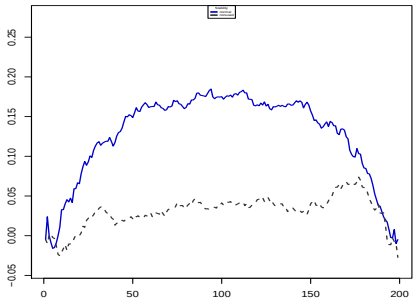
**(c)** ionosphere stability

**(d)** landsat train stability

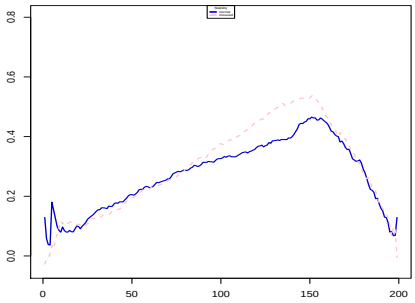
**(e)** leaf stability

**(f)** ma breast cancer stability

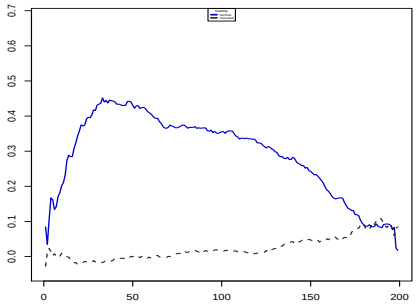
**Figure A.1:** Stability results for RSA SBG (1)



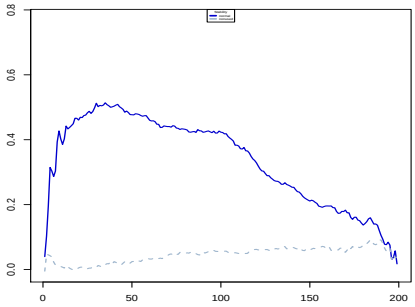
**(a)** ma colon tumor stability



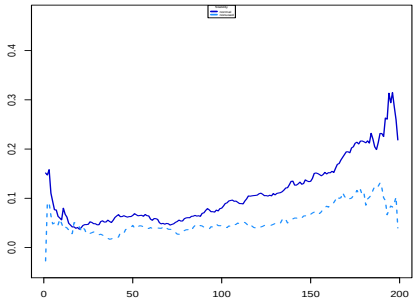
**(b)** ma gcm stability



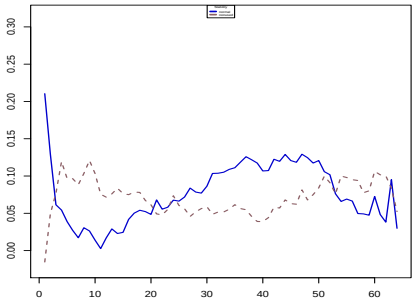
**(c)** ma leukemia stability



**(d)** ma lung cancer stability

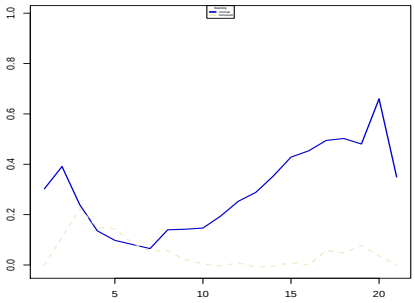


**(e)** ma prostate cancer stability

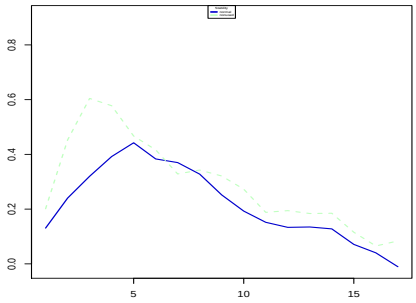


**(f)** mammogram stability

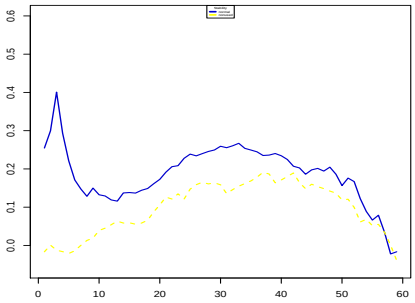
**Figure A.2:** Stability results for RSA SBG (2)



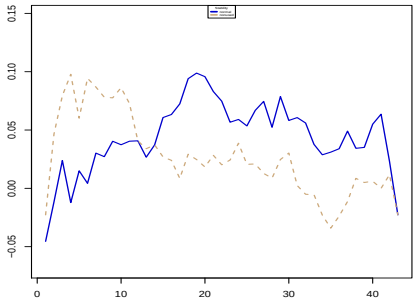
**(a)** parkinsons stability



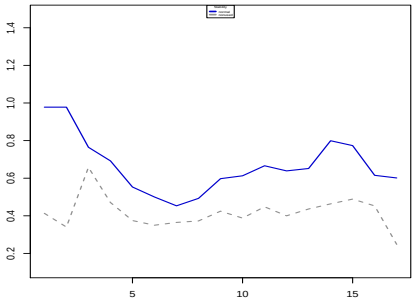
**(b)** pop failures stability



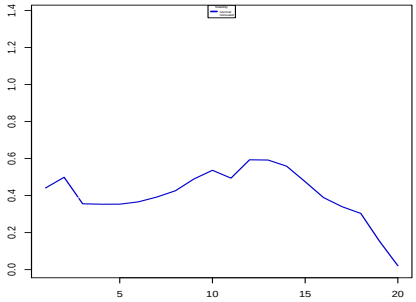
**(c)** sonar stability



**(d)** spectf stability

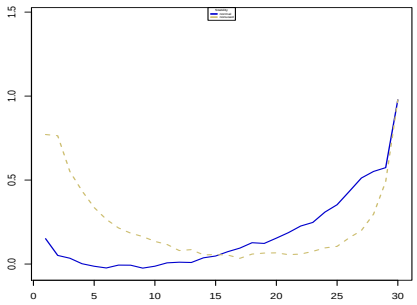


**(e)** vehicle stability

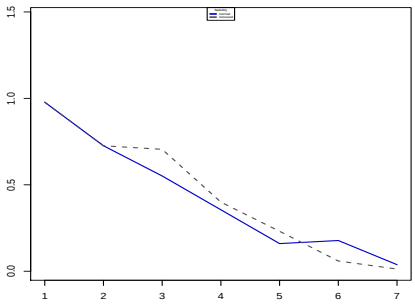


**(f)** waveform stability

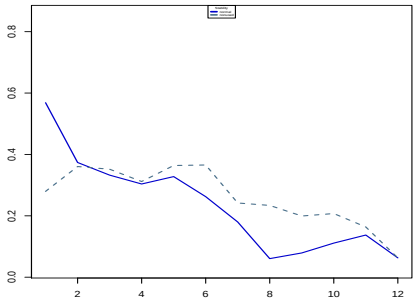
**Figure A.3:** Stability results for RSA SBG (3)



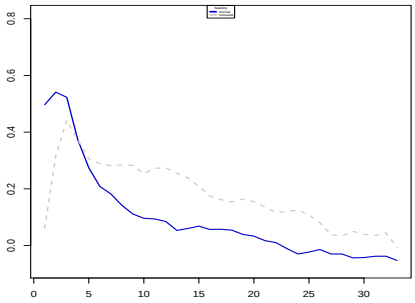
**(a)** wdbc stability



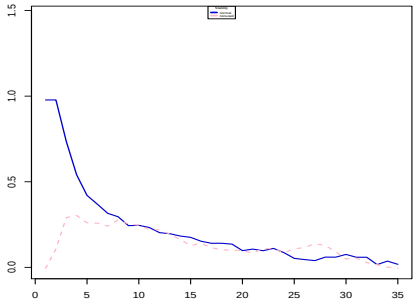
**(b)** diabetes stability



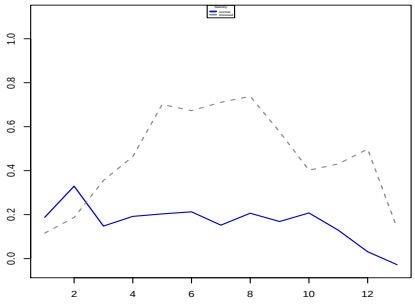
**(c)** heart-statlog stability



**(d)** ionosphere stability

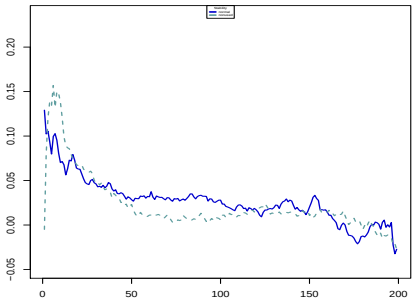


**(e)** landsat train stability

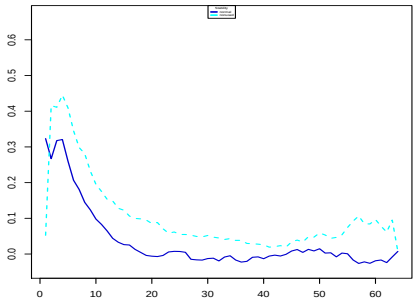


**(f)** leaf stability

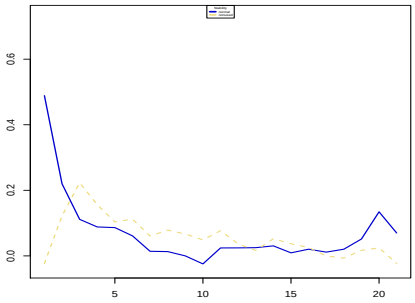
**Figure A.4:** Stability results for RSA SBG (4)



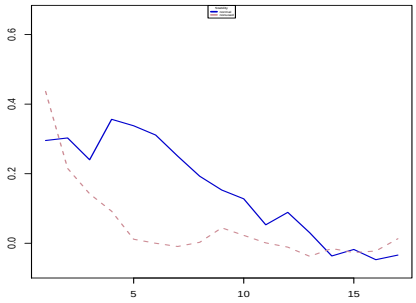
**(a)** ma prostate cancer stability



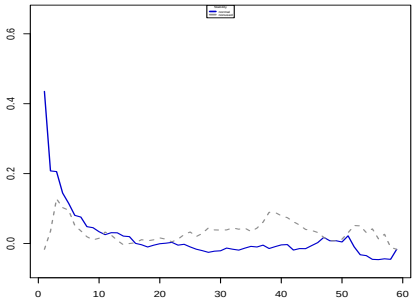
**(b)** mammogram stability



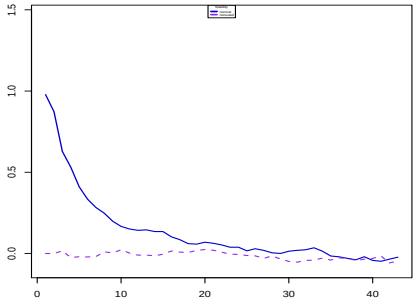
**(c)** parkinsons stability



**(d)** pop failures stability

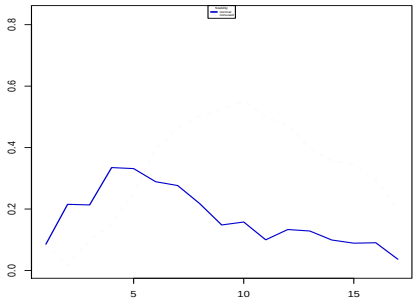


**(e)** sonar stability

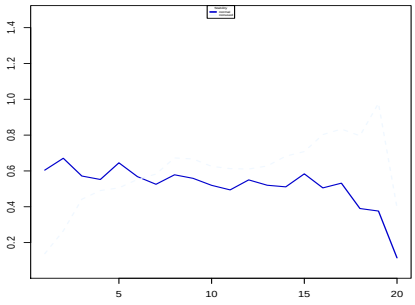


**(f)** spectf stability

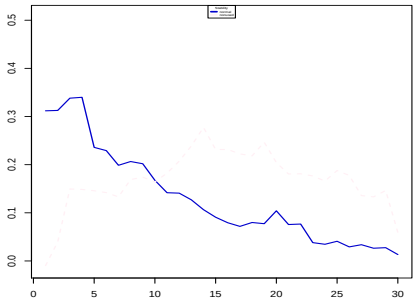
**Figure A.5:** Stability results for RSA SBG (5)



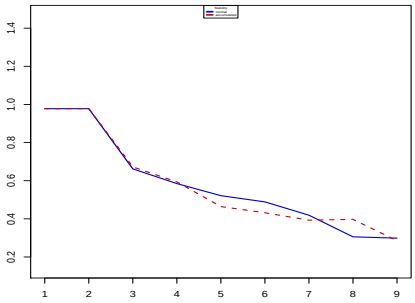
**(a)** vehicle stability



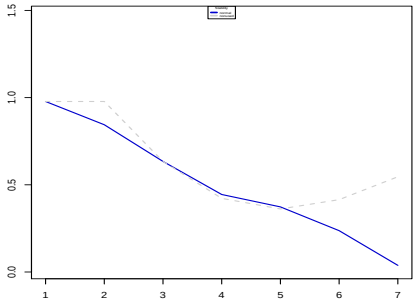
**(b)** waveform stability



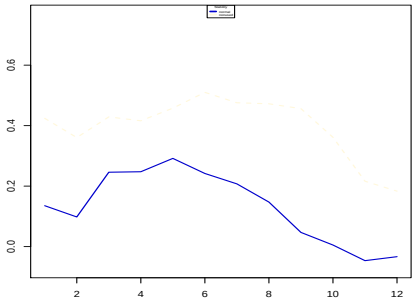
**(c)** wdbc stability



**(d)** antiCorrAl stability

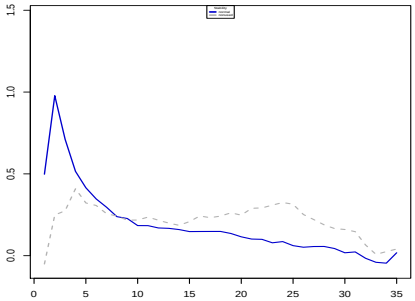


**(e)** diabetes stability

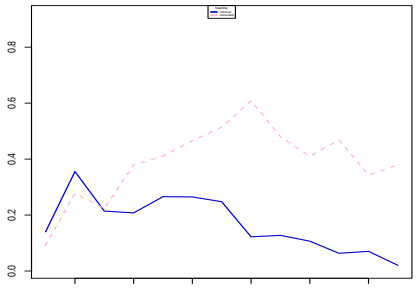


**(f)** heart-statlog stability

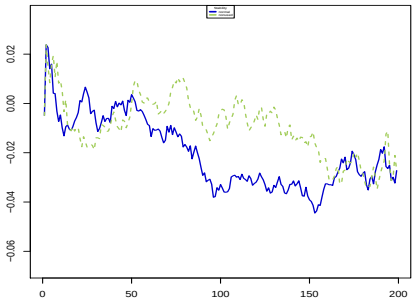
**Figure A.6:** Stability results for RSA SBG (6)



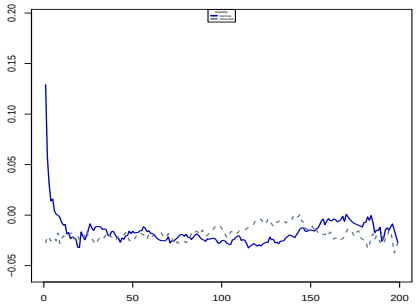
**(a)** landsat train stability



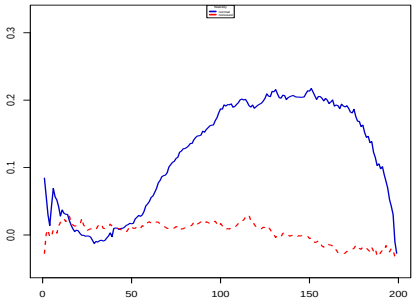
**(b)** leaf stability



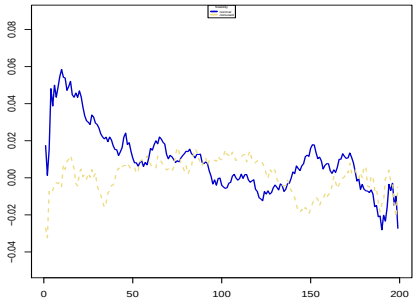
**(c)** ma breast cancer stability



**(d)** ma colon tumor stability

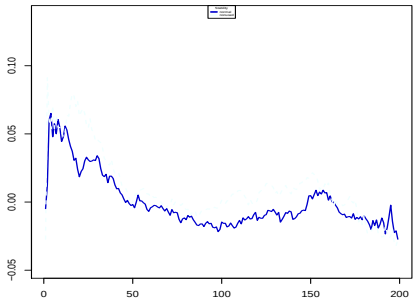


**(e)** ma leukemia stability

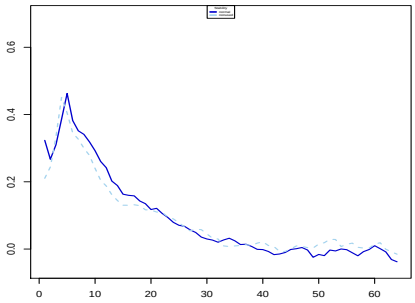


**(f)** ma lung cancer stability

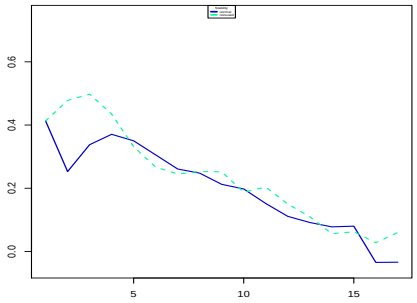
**Figure A.7:** Stability results for RSA SBG (7)



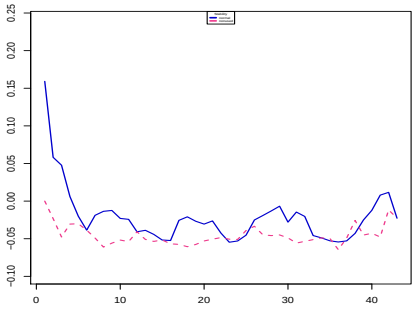

**(a)** ma prostate cancer stability



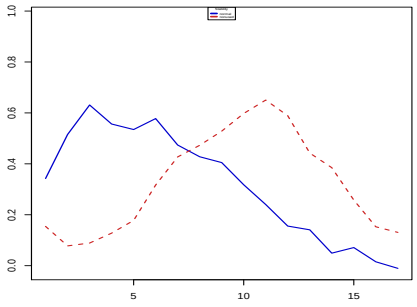

**(b)** mammogram stability



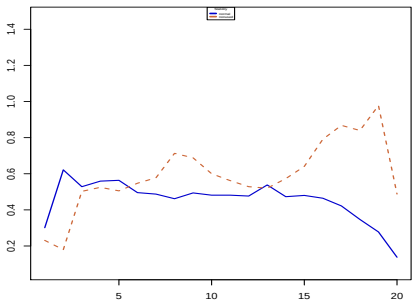

**(c)** pop failures stability




**(d)** spectf stability

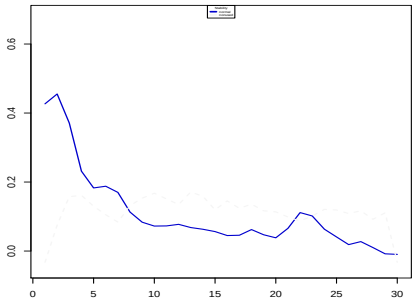



**(e)** vehicle stability



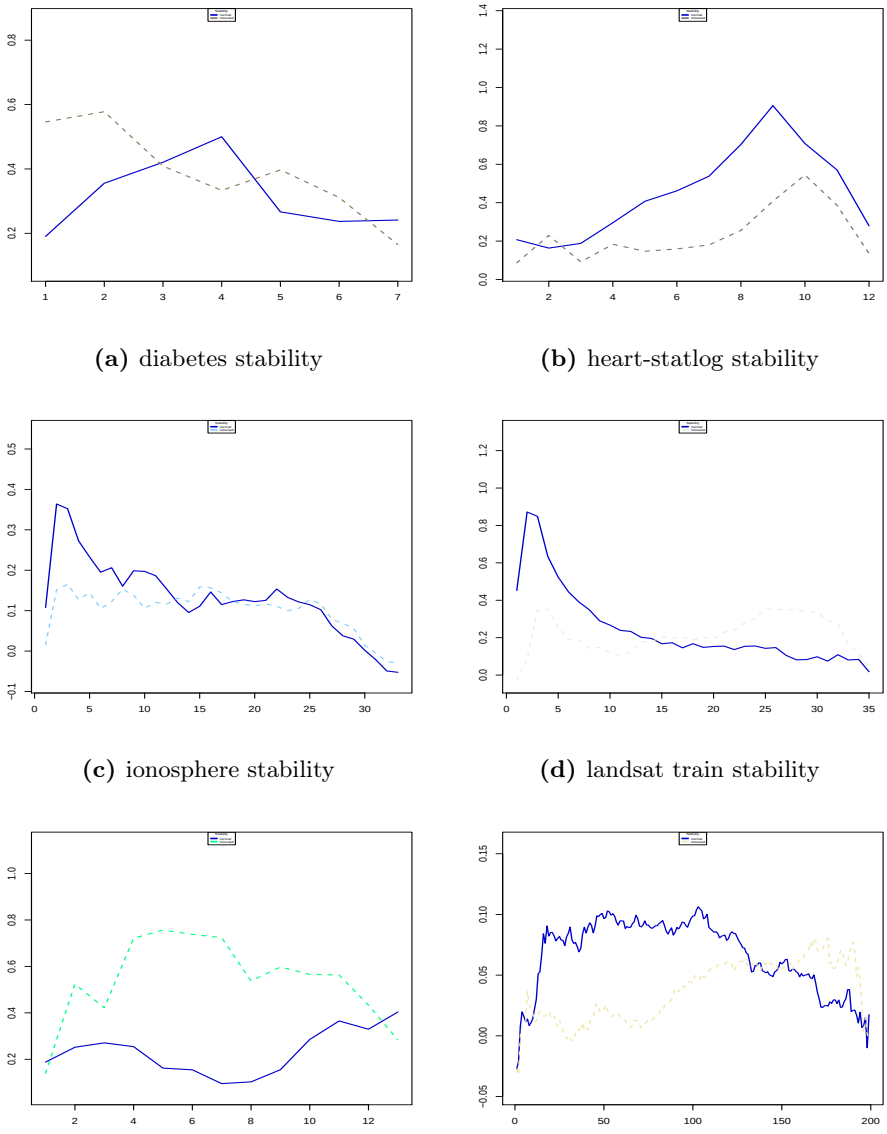

**(f)** waveform stability

**Figure A.8:** Stability results for RSA SBG (8)




**(a)** wdbc stability

**Figure A.9:** Stability results for RSA SFG (9)



**(a)** diabetes stability

**(b)** heart-statlog stability

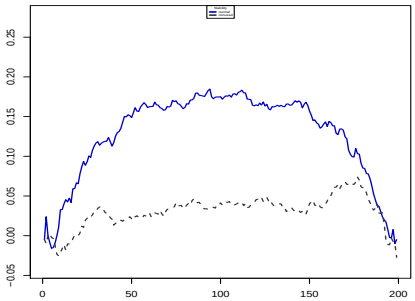
**(c)** ionosphere stability

**(d)** landsat train stability

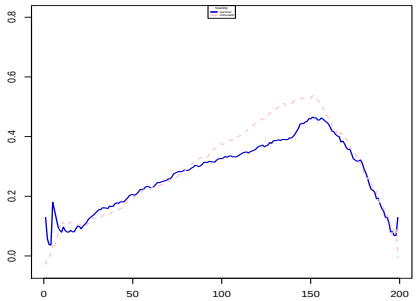
**(e)** leaf stability

**(f)** ma breast cancer stability

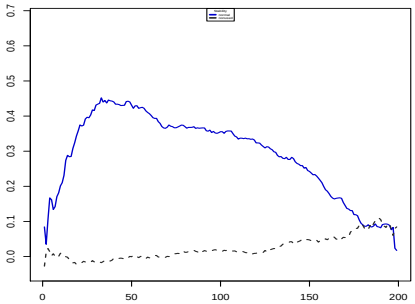
**Figure A.10:** Stability results for RSA SBG (10)



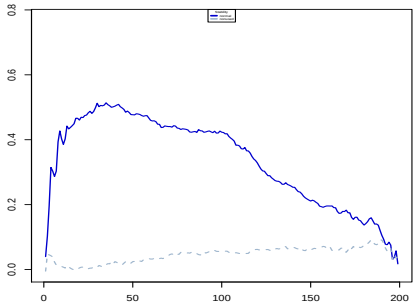
**(a)** ma colon tumor stability



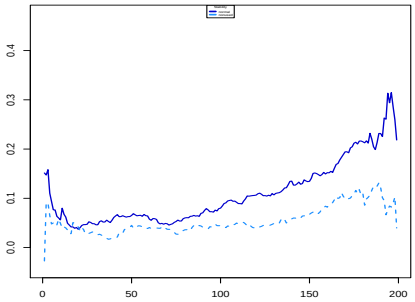
**(b)** ma gcm stability



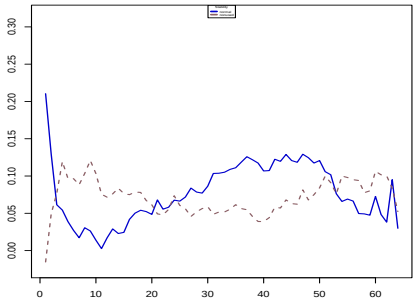
**(c)** ma leukemia stability



**(d)** ma lung cancer stability

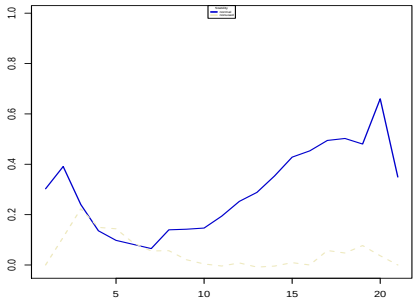


**(e)** ma prostate cancer stability

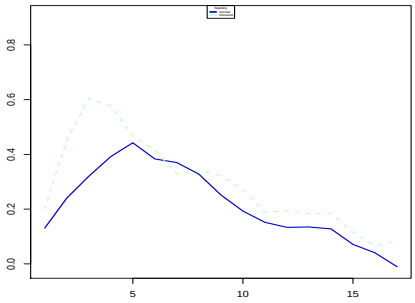


**(f)** mammogram stability

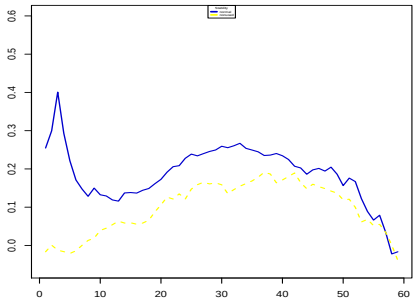
**Figure A.11:** Stability results for RSA SBG (11)



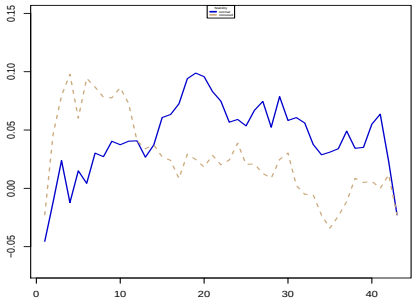
**(a)** parkinsons stability



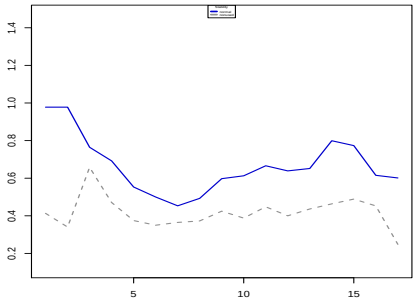
**(b)** pop failures stability



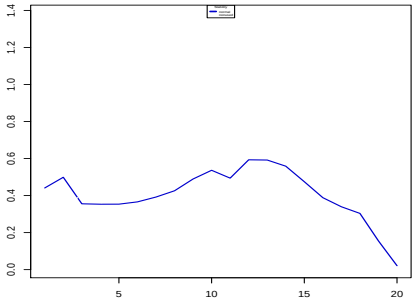
**(c)** sonar stability



**(d)** spectf stability

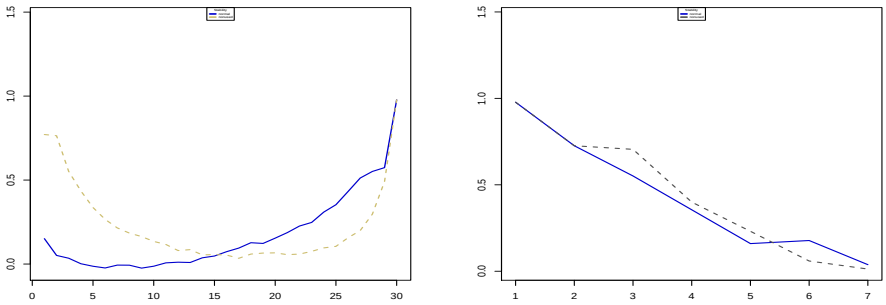


**(e)** vehicle stability



**(f)** waveform stability

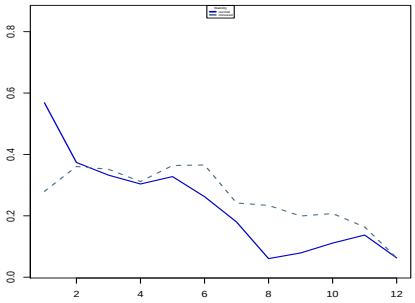
**Figure A.12:** Stability results for RSA SBG (12)



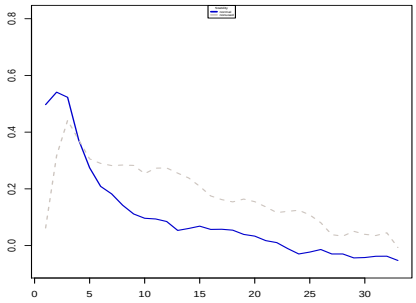
**(a)** wdbc stability



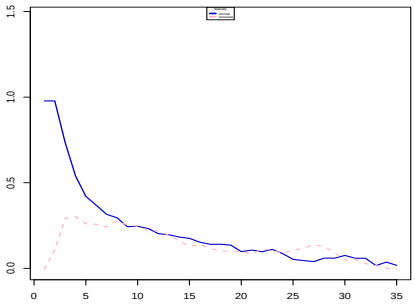
**(b)** diabetes stability



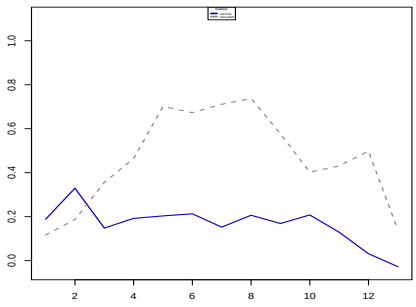
**(c)** heart-statlog stability



**(d)** ionosphere stability

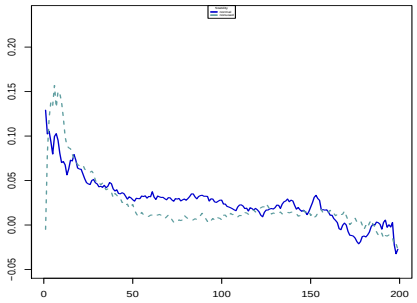


**(e)** landsat train stability

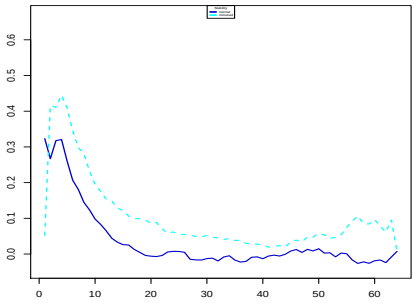


**(f)** leaf stability

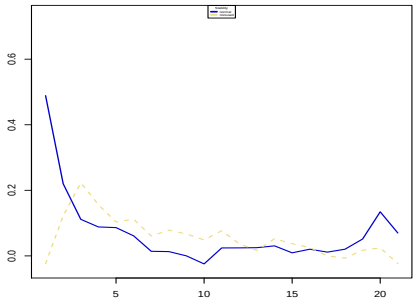
**Figure A.13:** Stability results for RSA SBG (13)



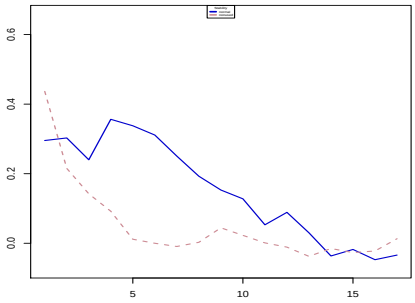
**(a)** ma prostate cancer stability



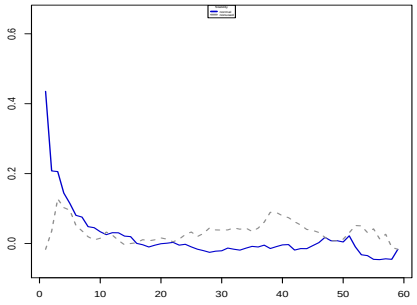
**(b)** mammogram stability



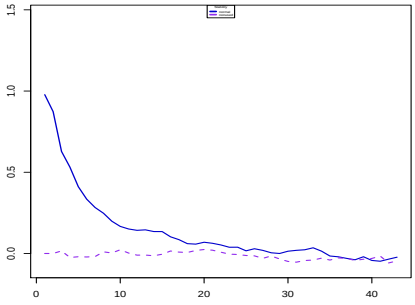
**(c)** parkinsons stability



**(d)** pop failures stability

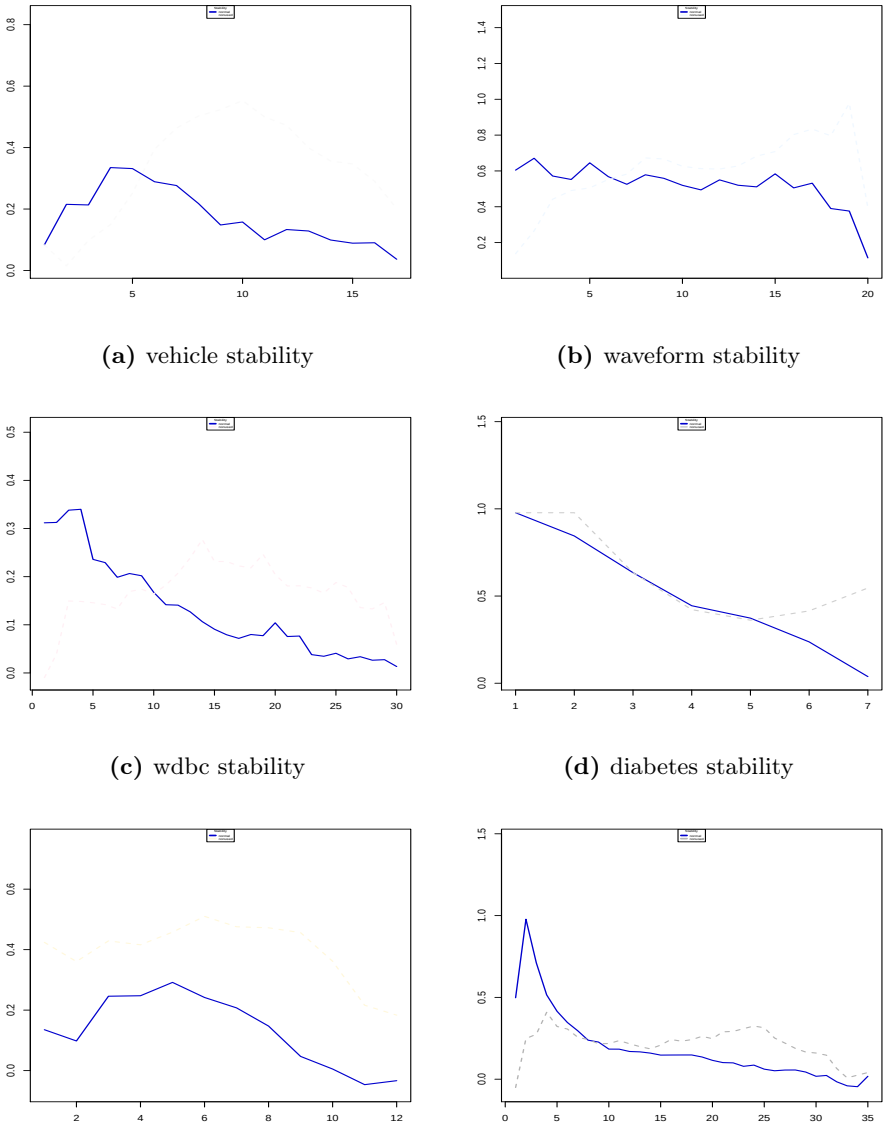


**(e)** sonar stability



**(f)** spectf stability

**Figure A.14:** Stability results for RSA SBG (14)



**(a)** vehicle stability

**(b)** waveform stability

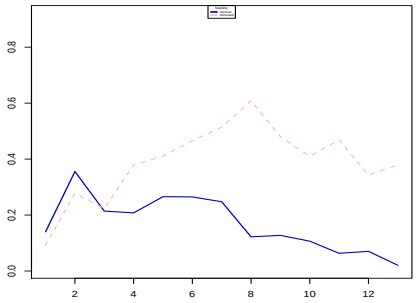
**(c)** wdbc stability

**(d)** diabetes stability

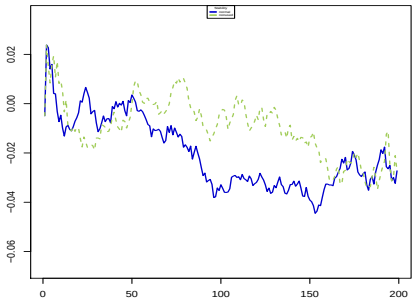
**(e)** heart-statlog stability

**(f)** landsat train stability

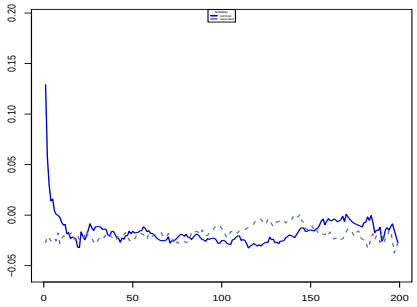
**Figure A.15:** Stability results for RSA SBG (15)



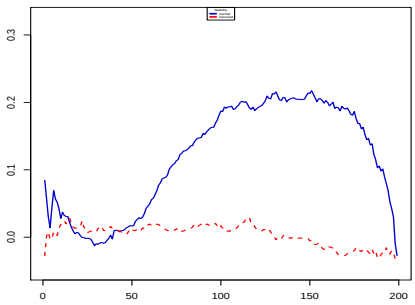

**(a)** leaf stability



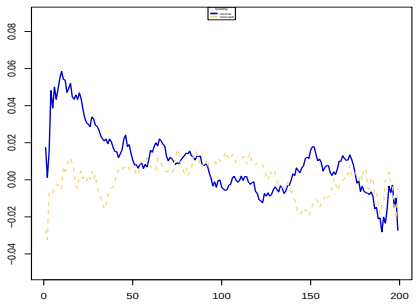

**(b)** ma breast cancer stability



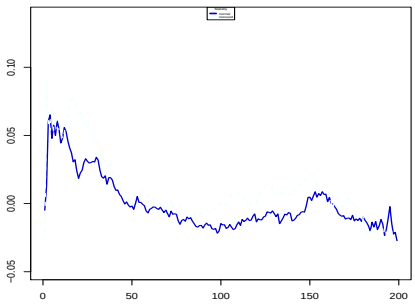

**(c)** ma colon tumor stability




**(d)** ma leukemia stability

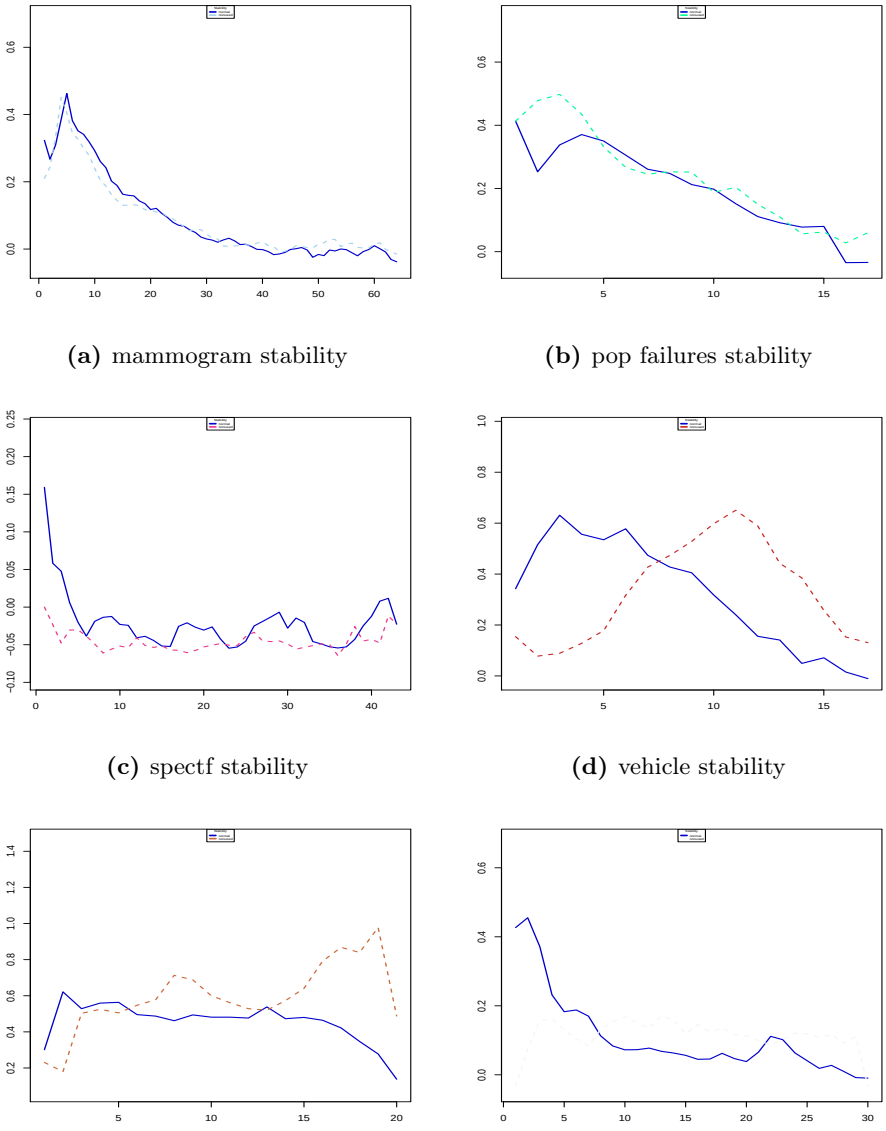



**(e)** ma lung cancer stability




**(f)** ma prostate cancer stability

**Figure A.16:** Stability results for RSA SBG (16)



**(a)** mammogram stability

**(b)** pop failures stability

**(c)** spectf stability

**(d)** vehicle stability

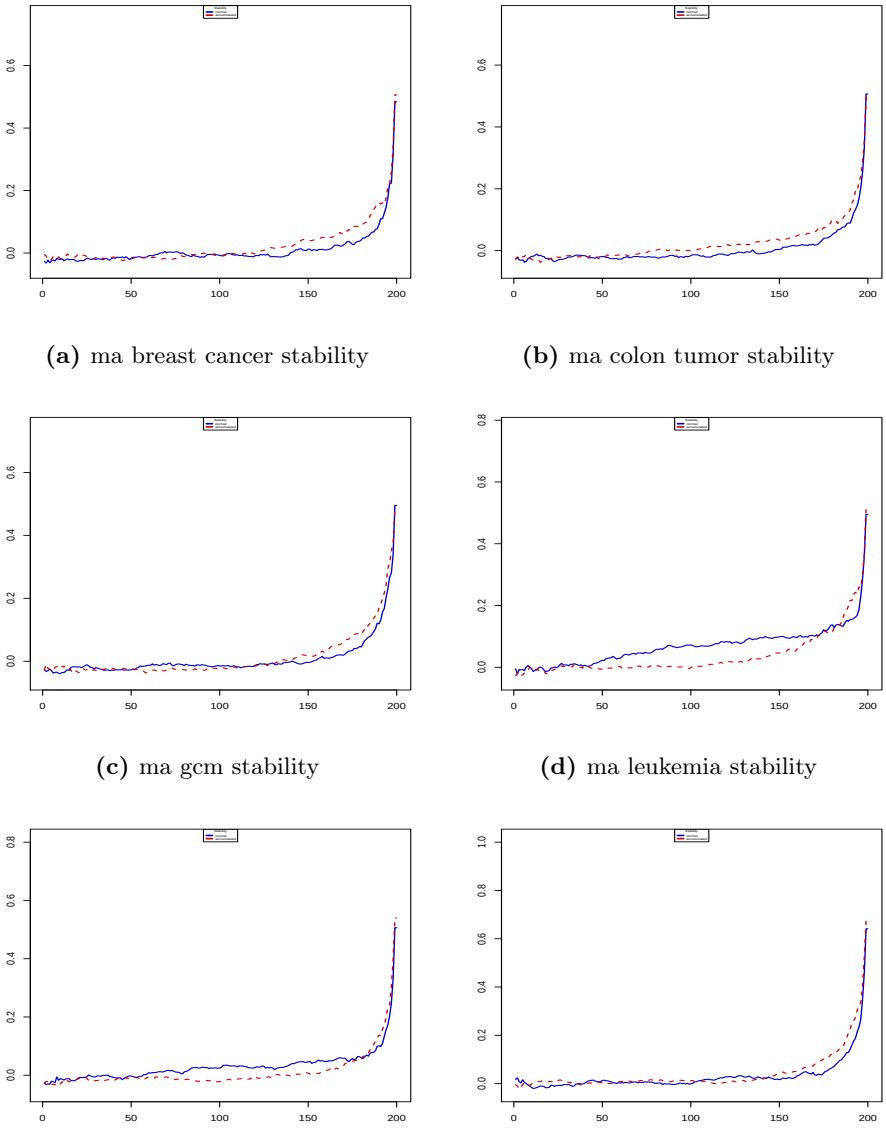
**(e)** waveform stability

**(f)** wdbc stability

**Figure A.17:** Stability results for RSA SBG (17)

Appendix

# B Accumulated evidence detailed results



**(a)** ma breast cancer stability

**(b)** ma colon tumor stability

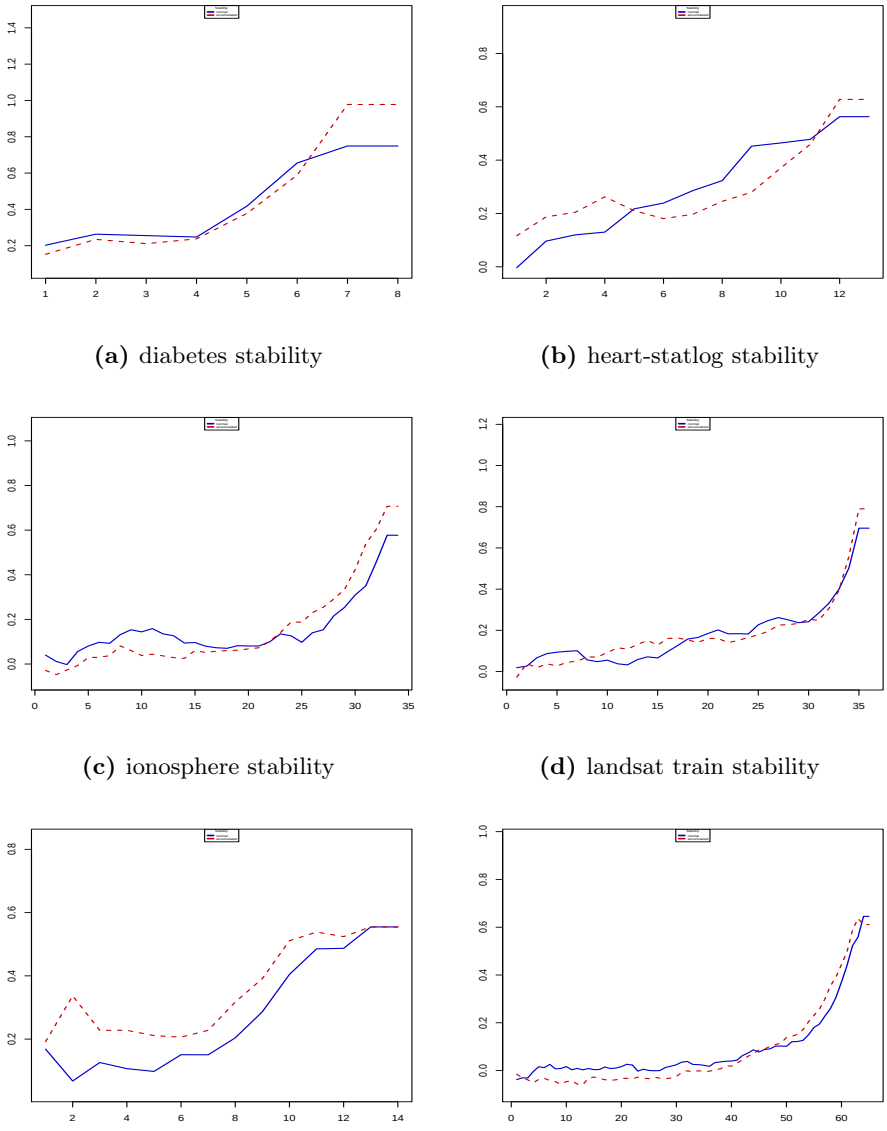
**(c)** ma gcm stability

**(d)** ma leukemia stability

**(e)** ma lung cancer stability

**(f)** ma prostate cancer stability

**Figure B.1:** Stability results for $SBG^{+}$ vs SBG (1)



**(a)** diabetes stability

**(b)** heart-statlog stability

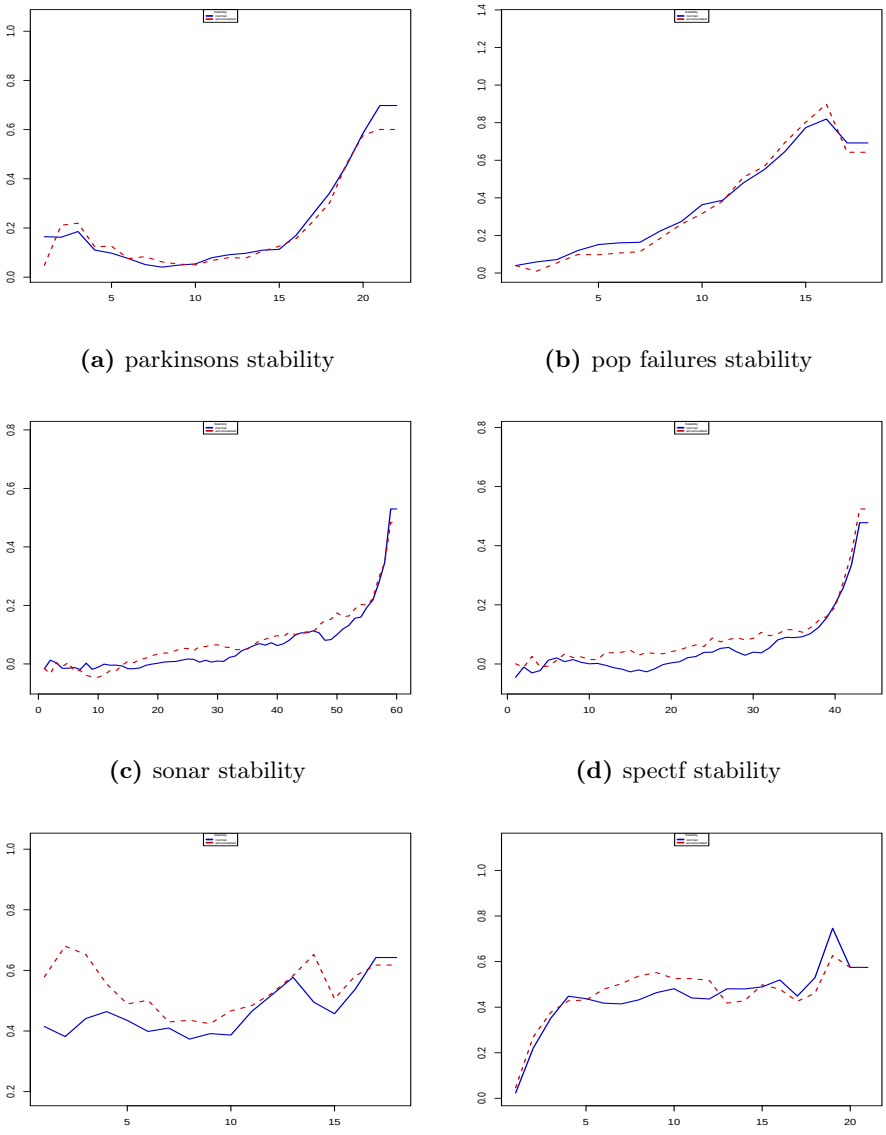
**(c)** ionosphere stability

**(d)** landsat train stability

**(e)** leaf stability

**(f)** mammogram stability

**Figure B.2:** Stability results for $SBG^{+}$ vs SBG (2)



**(a)** parkinsons stability

**(b)** pop failures stability

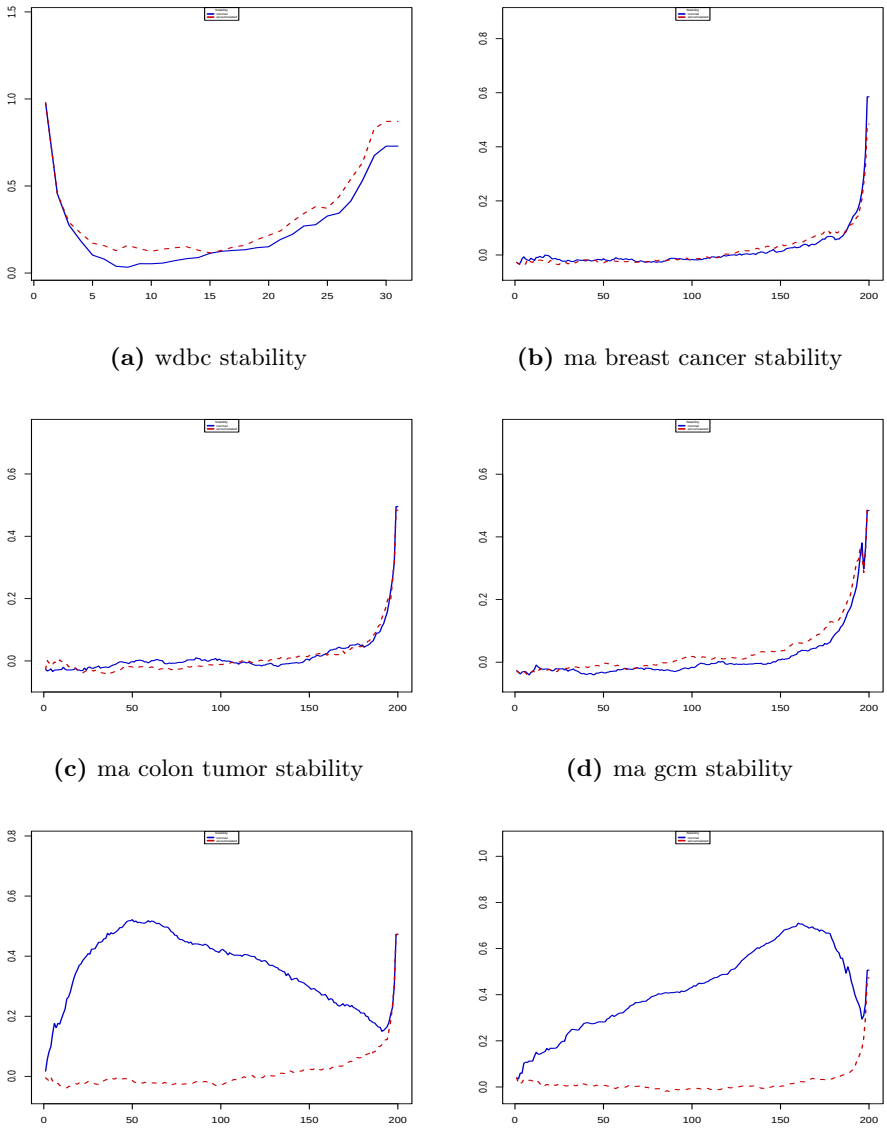
**(c)** sonar stability

**(d)** spectf stability

**(e)** vehicle stability

**(f)** waveform stability

**Figure B.3:** Stability results for $SBG^{+}$ vs SBG (3)



**(a)** wdbc stability

**(b)** ma breast cancer stability

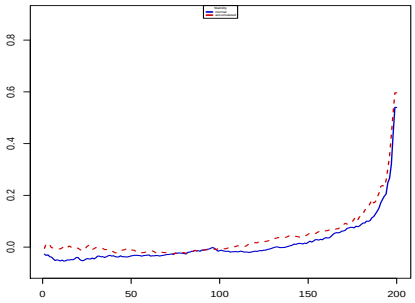
**(c)** ma colon tumor stability

**(d)** ma gcm stability

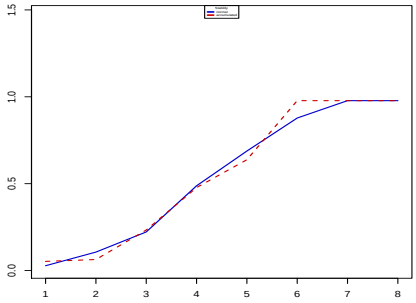
**(e)** ma leukemia stability

**(f)** ma lung cancer stability

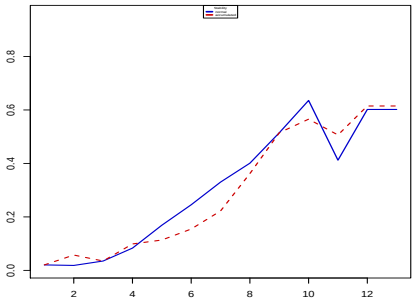
**Figure B.4:** Stability results for $SBG^{+}$ vs SBG (4)



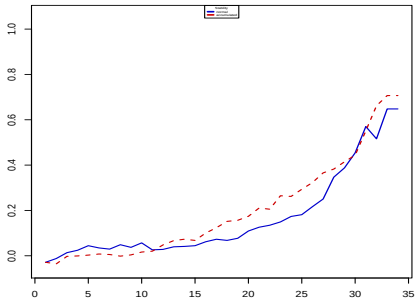
**(a)** ma prostate cancer stability



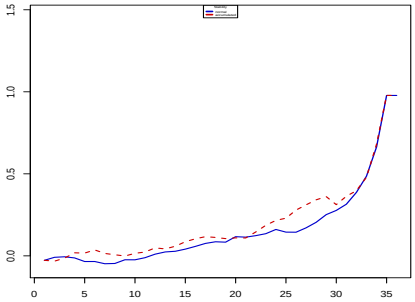
**(b)** diabetes stability



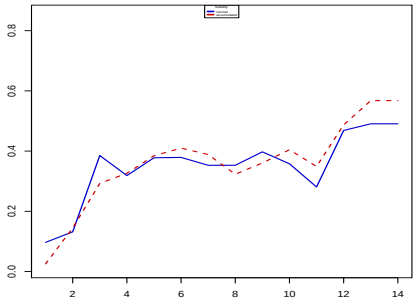
**(c)** heart-statlog stability



**(d)** ionosphere stability

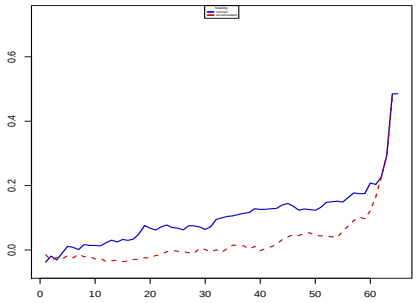


**(e)** landsat train stability

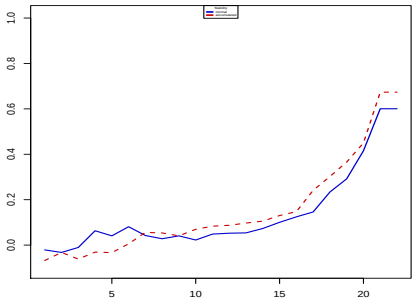


**(f)** leaf stability

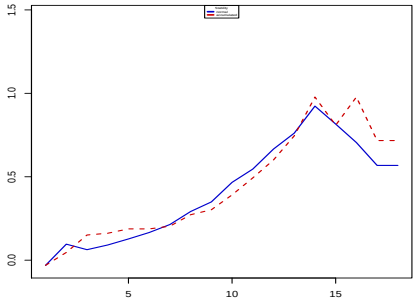
**Figure B.5:** Stability results for $SBG^{+}$ vs SBG (5)



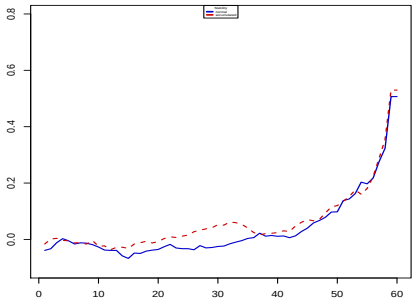
**(a)** mammogram stability



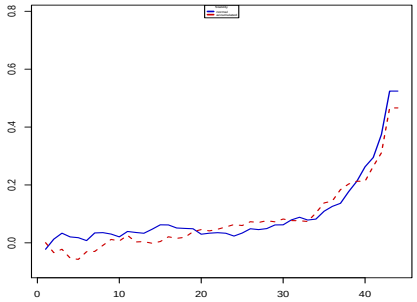
**(b)** parkinsons stability



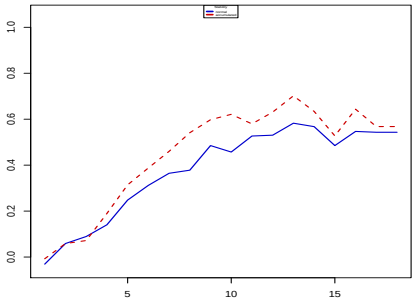
**(c)** pop failures stability



**(d)** sonar stability

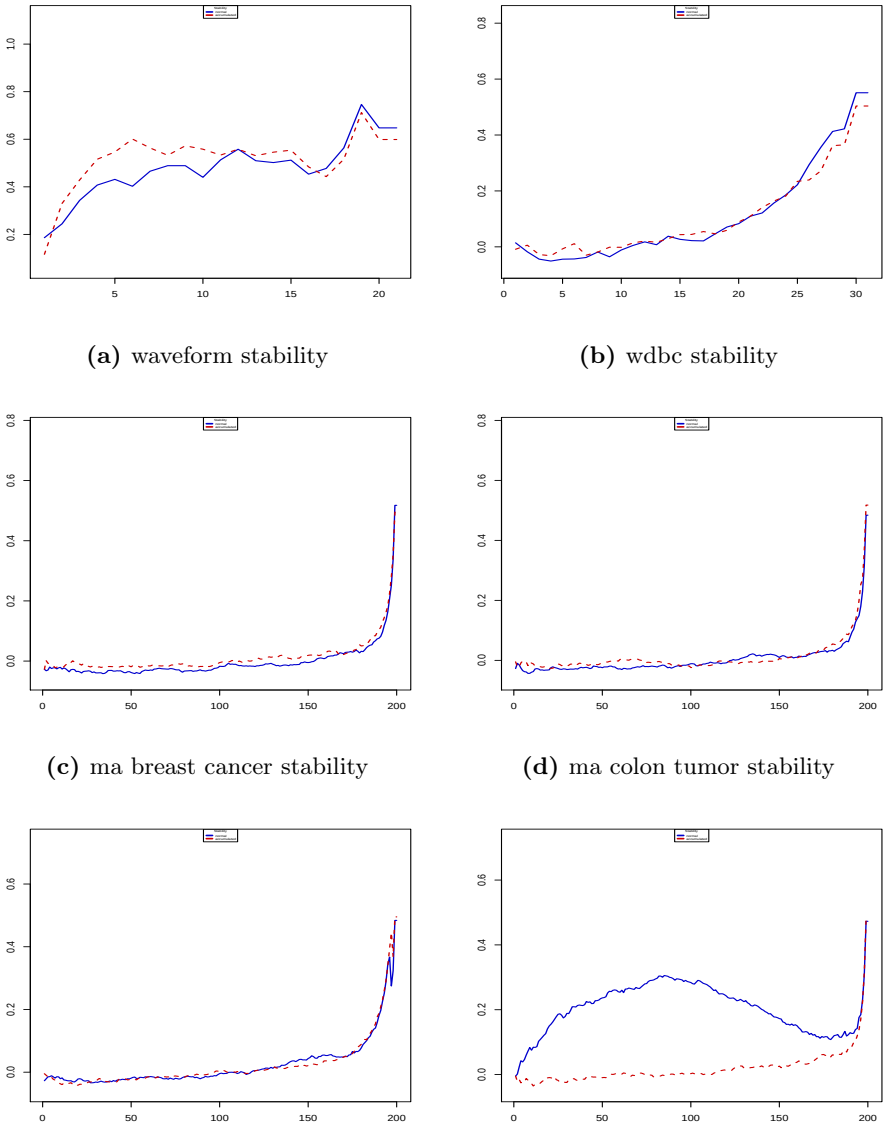


**(e)** spectf stability



**(f)** vehicle stability

**Figure B.6:** Stability results for $SBG^{+}$ vs SBG (6)



**(a)** waveform stability

**(b)** wdbc stability

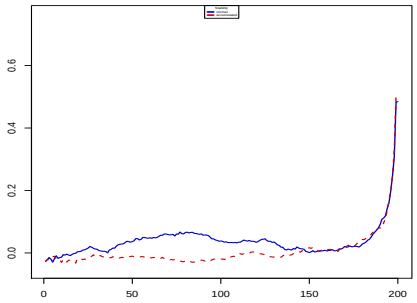
**(c)** ma breast cancer stability

**(d)** ma colon tumor stability

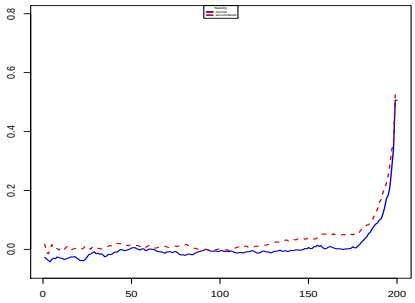
**(e)** ma gcm stability

**(f)** ma leukemia stability

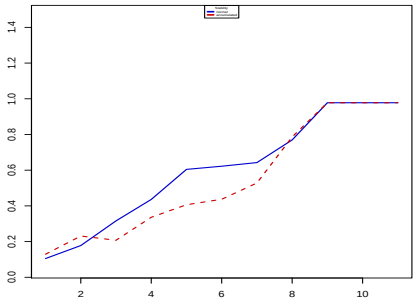
**Figure B.7:** Stability results for $SBG^{+}$ vs SBG (7)



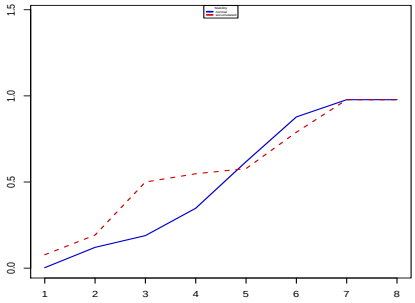
**(a)** ma lung cancer stability



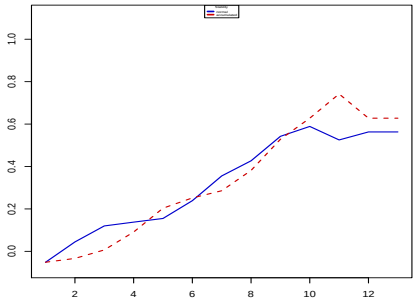
**(b)** ma prostate cancer stability



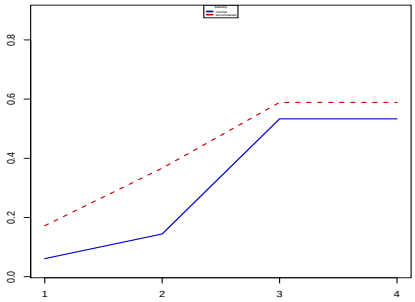
**(c)** antiCorrAl stability



**(d)** diabetes stability

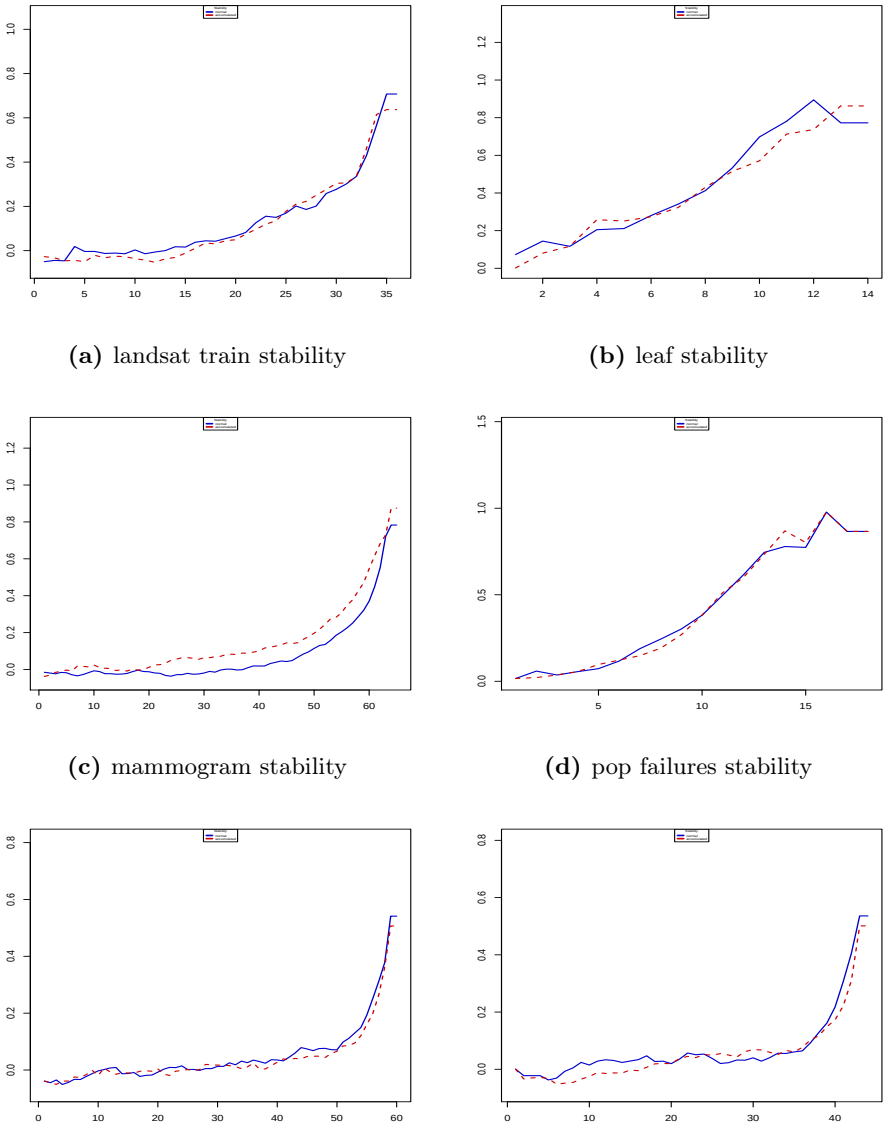


**(e)** heart-statlog stability



**(f)** iris stability

**Figure B.8:** Stability results for $SBG^{+}$ vs SBG (8)



**(a)** landsat train stability

**(b)** leaf stability

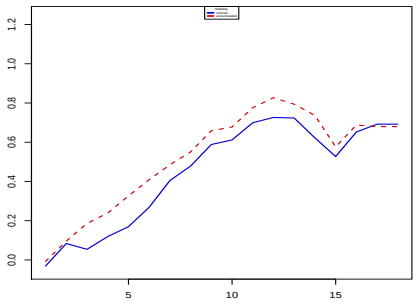
**(c)** mammogram stability

**(d)** pop failures stability

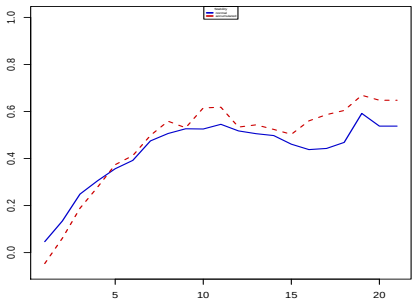
**(e)** sonar stability

**(f)** spectf stability

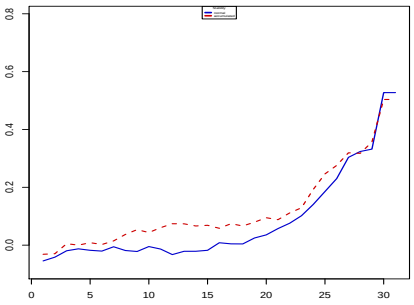
**Figure B.9:** Stability results for $SBG^{+}$ vs SBG (9)




**(a)** vehicle stability




**(b)** waveform stability

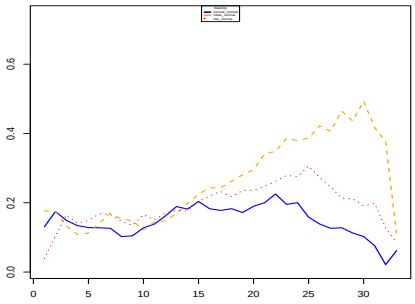



**(c)** wdbc stability

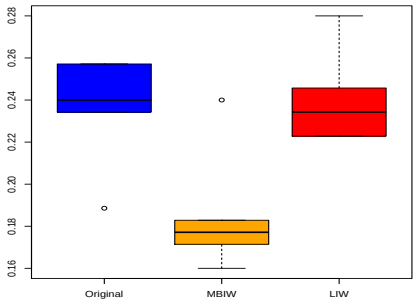
**Figure B.10:** Stability results for $SBG^{+}$ vs SBG (10)

Appendix

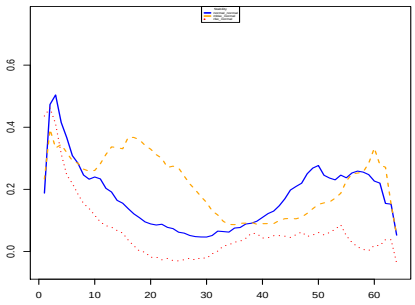
# C Instance and feature weighing combination detailed results



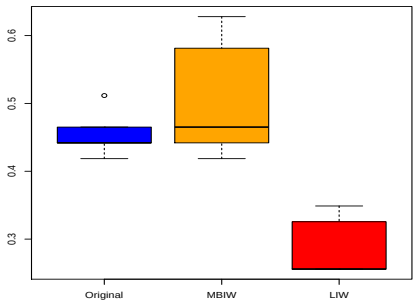
**(a)** ionosphere stability



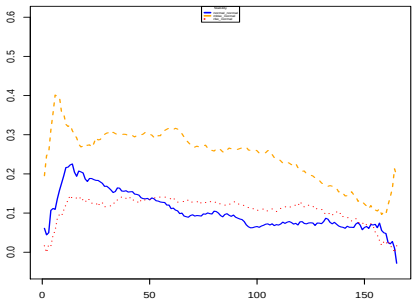

**(b)** ionosphere error



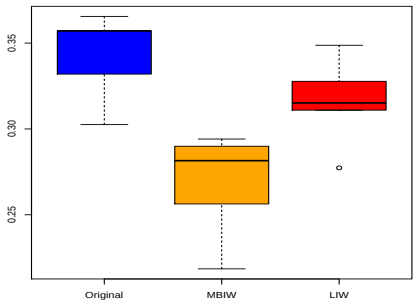
**(c)** mammogram stability




**(d)** mammogram error

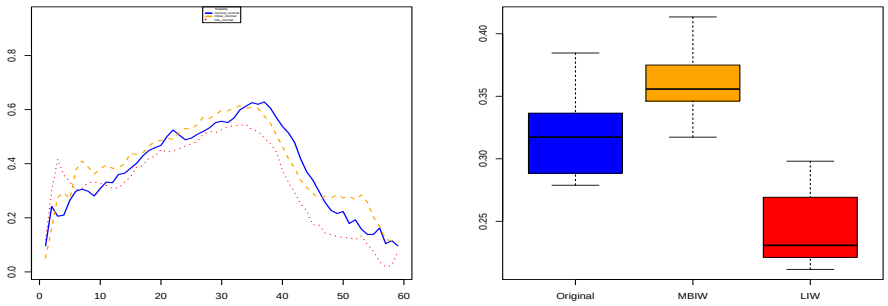


**(e)** musk stability

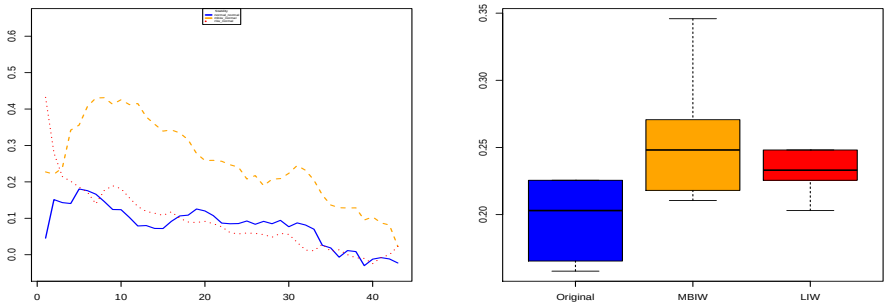



**(f)** musk error

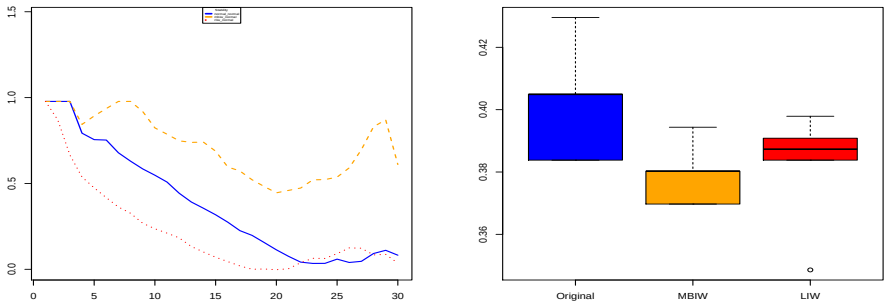
**Figure C.1:** Feature stability on UCI data with $normal\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** sonar stability **(b)** sonar error

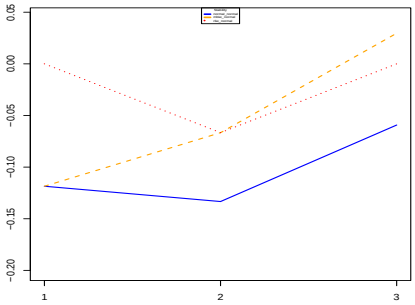



**(c)** spectf stability **(d)** spectf error

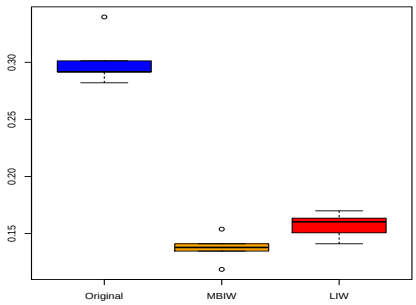



**(e)** wdbc stability **(f)** wdbc error

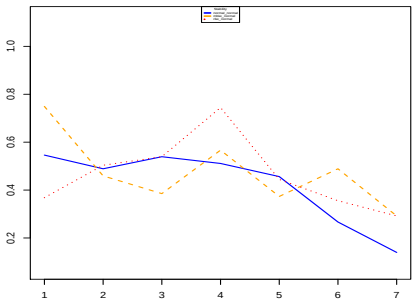
**Figure C.2:** Feature stability on UCI data with $normal\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



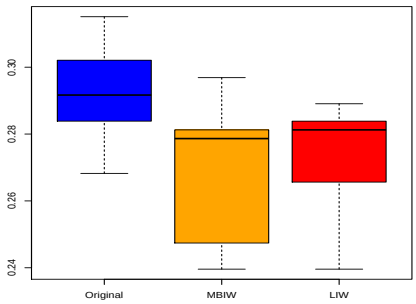
**(a)** balance scale stability



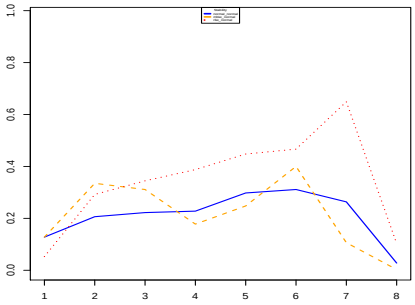

**(b)** balance scale error



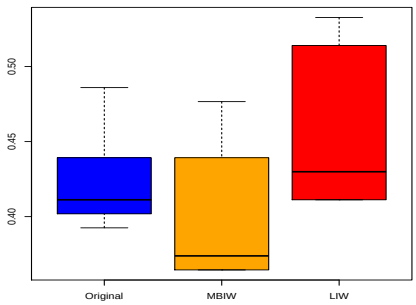
**(c)** diabetes stability




**(d)** diabetes error

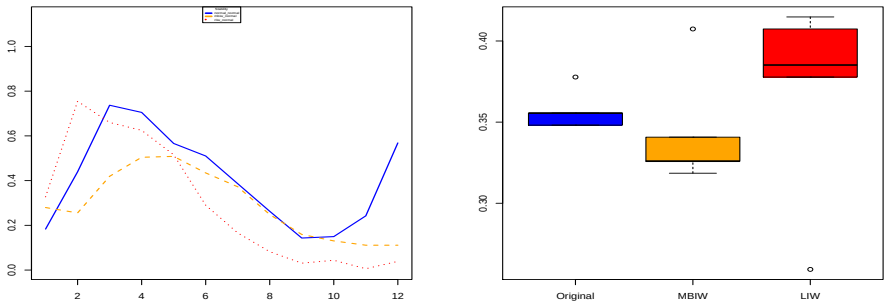


**(e)** glass stability

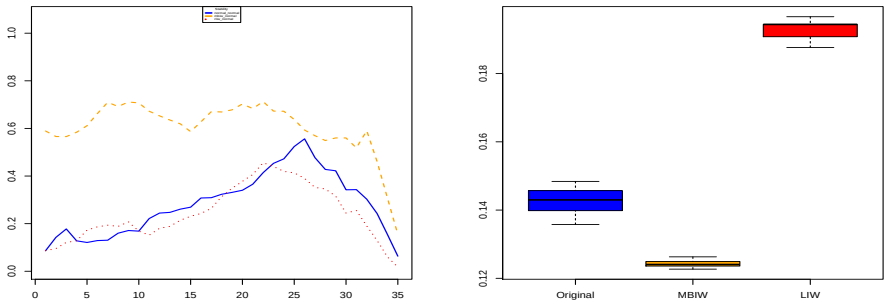



**(f)** glass error

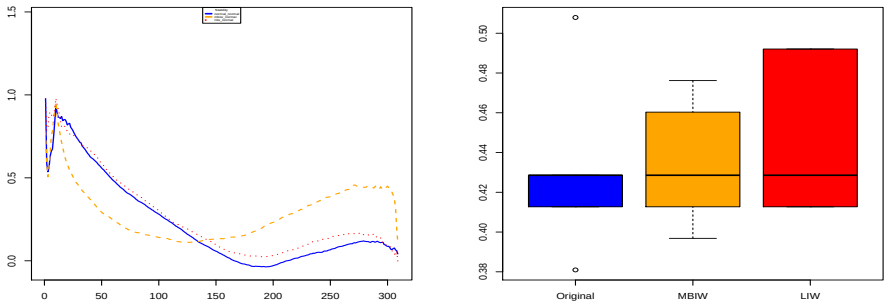
**Figure C.3:** Feature stability on UCI data with $normal\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** heart statlog stability **(b)** heart statlog error

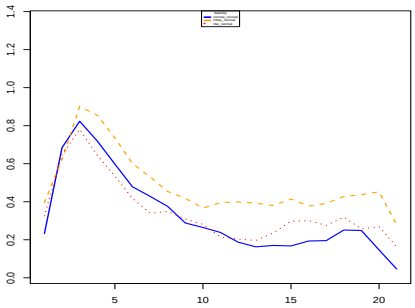



**(c)** landsat stability **(d)** landsat error

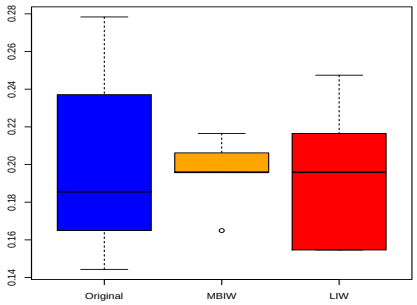



**(e)** lsvt voice stability **(f)** lsvt voice error

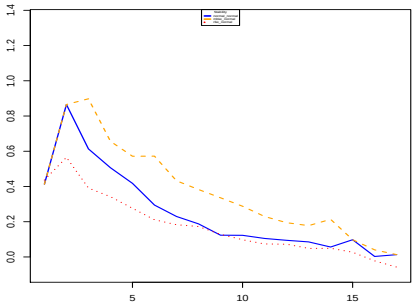
**Figure C.4:** Feature stability on UCI data with $normal\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



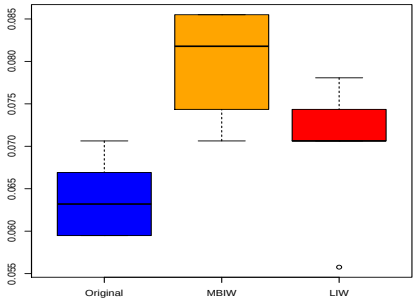
**(a)** parkinsons statlog stability



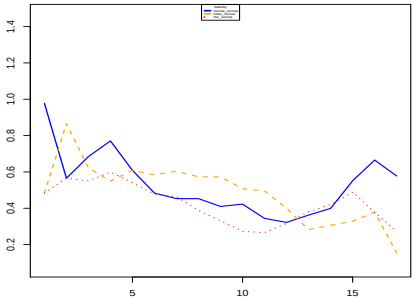

**(b)** parkinsons statlog error



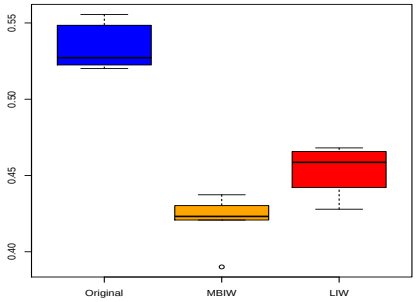
**(c)** pop failures stability




**(d)** pop failures error

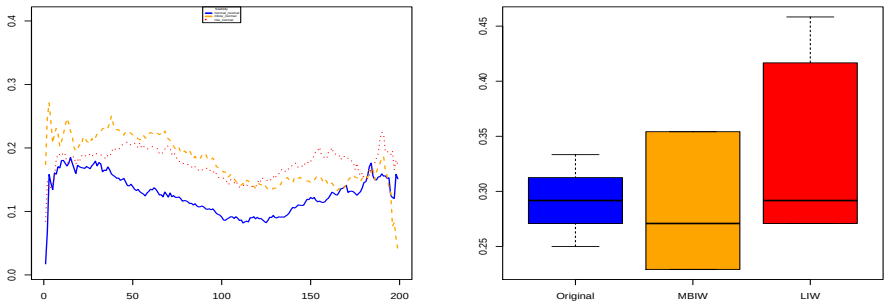


**(e)** vehicle stability




**(f)** vechicle voice error

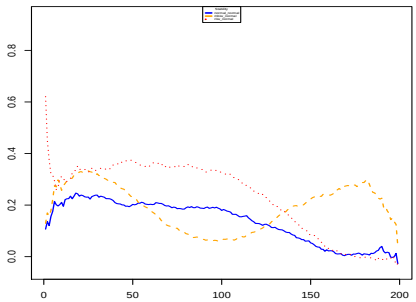
**Figure C.5:** Feature stability on UCI data with *normal*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



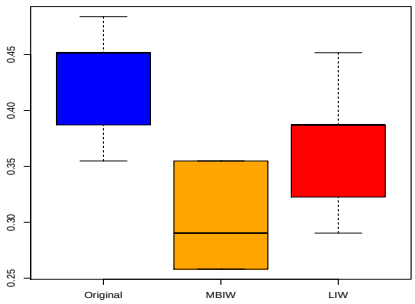

**(a)** breast cancer stability



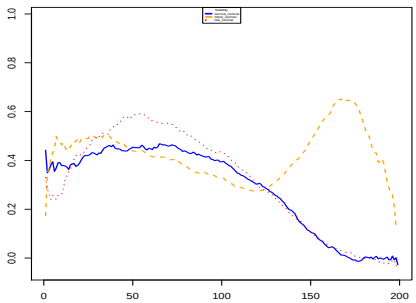

**(b)** breast cancer error



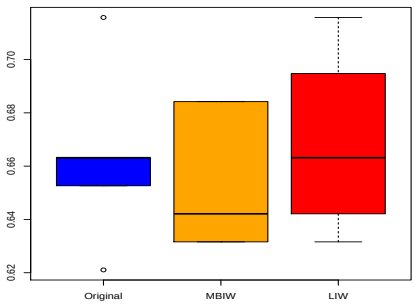

**(c)** colon tumor stability




**(d)** colon tumor error

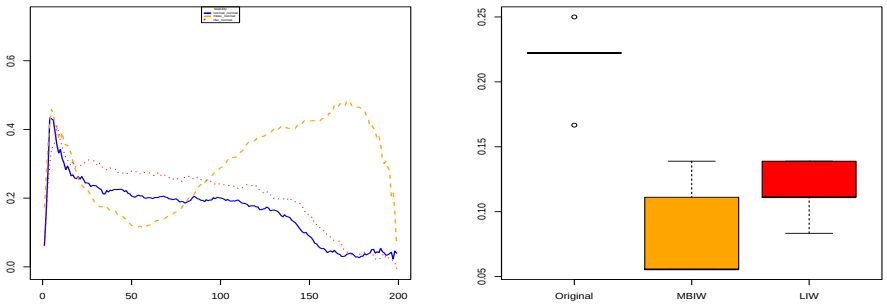



**(e)** gcm stability

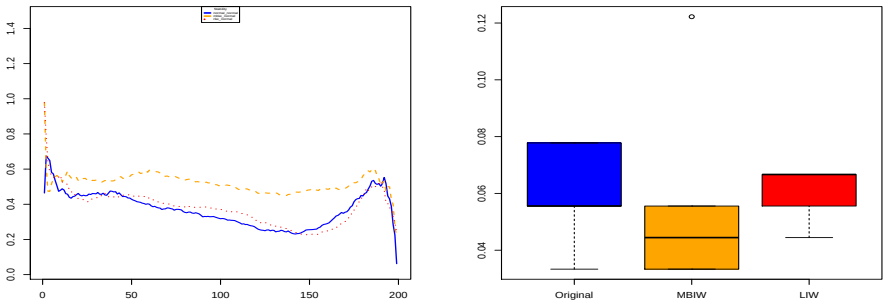



**(f)** gcm error

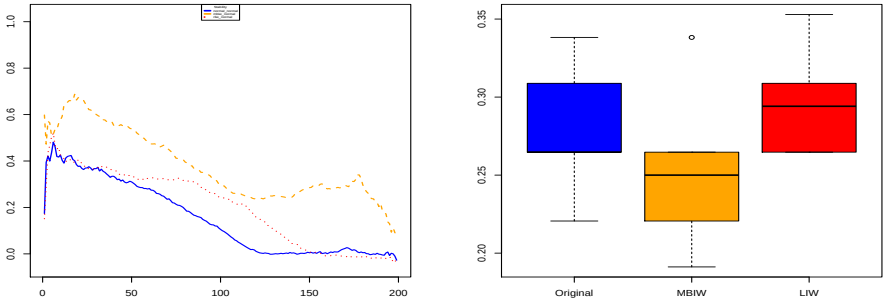
**Figure C.6:** Feature stability on microarray data with *normal*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** version). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** leukemia stability **(b)** leukemia error

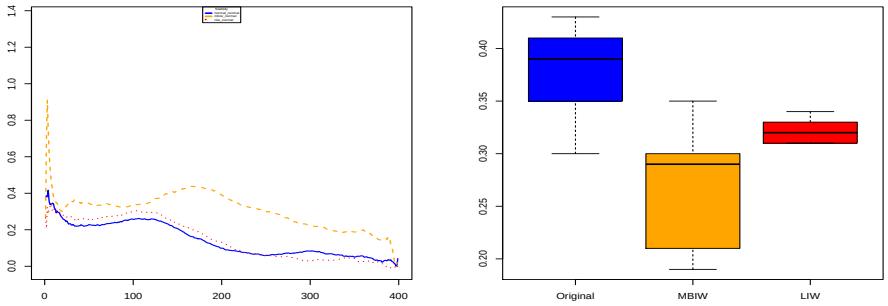



**(c)** lung cancer stability **(d)** lung cancer error

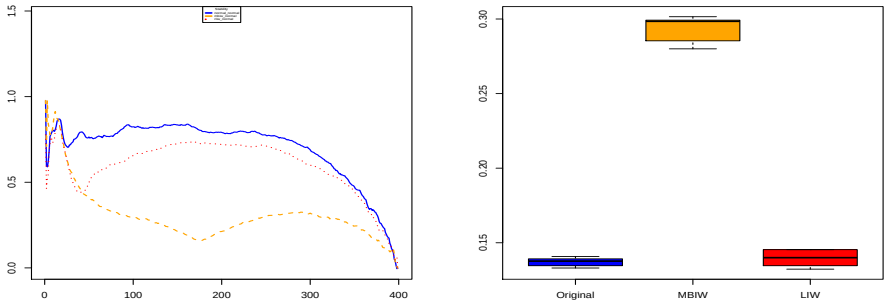



**(e)** prostate cancer stability **(f)** prostate cancer error

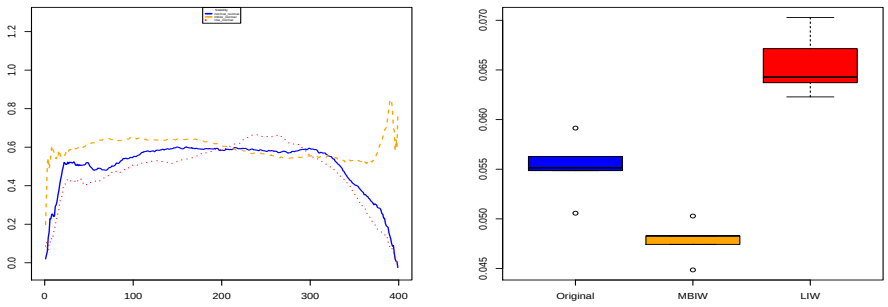
**Figure C.7:** Feature stability on microarray data with *normal*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** arcene stability **(b)** arcene error

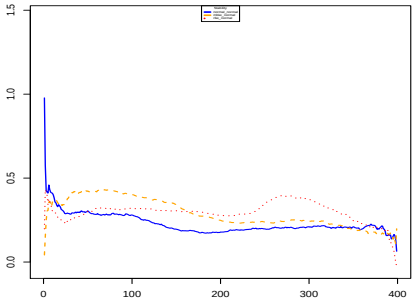



**(c)** madelon stability **(d)** madelon error

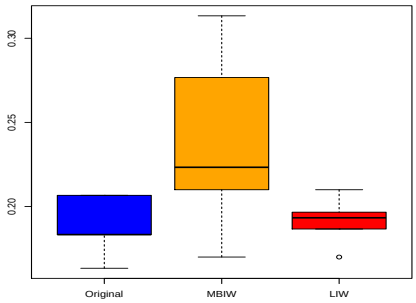



**(e)** gisette stability **(f)** gisette error

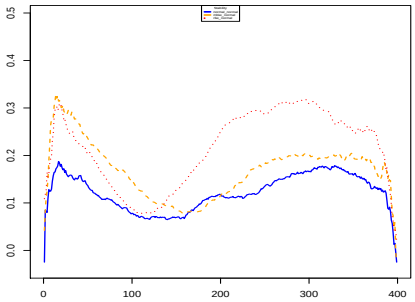
**Figure C.8:** Feature stability on NIPS Challenge data with $normal\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



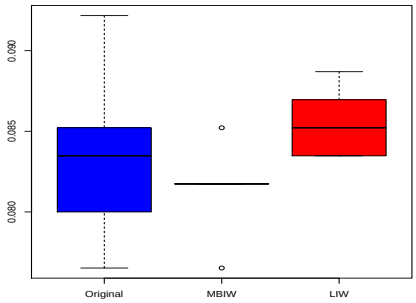

**(a)** dexter stability




**(b)** dexter error

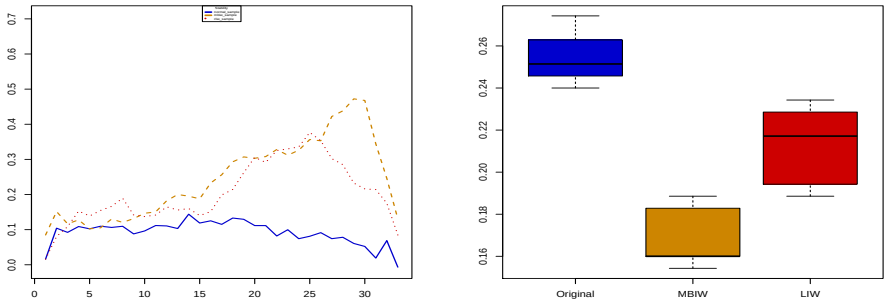



**(c)** dorothea stability

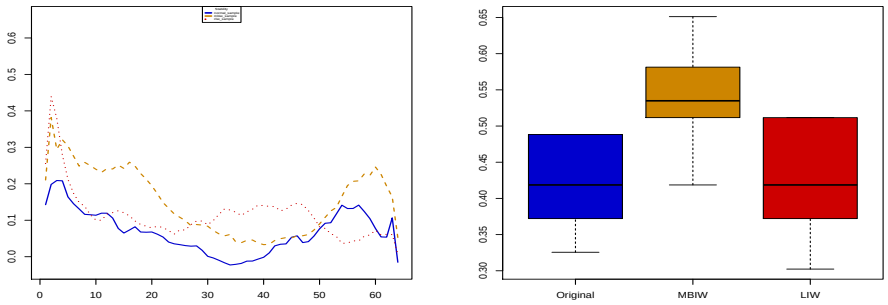



**(d)** dorothea error

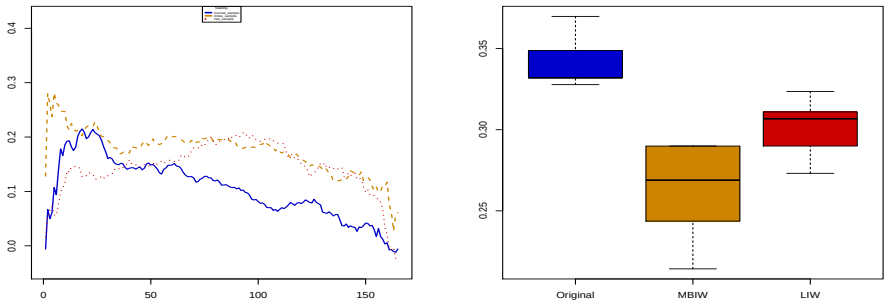
**Figure C.9:** Feature stability on NIPS Challenge data with $normal\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** ionosphere stability **(b)** ionosphere error

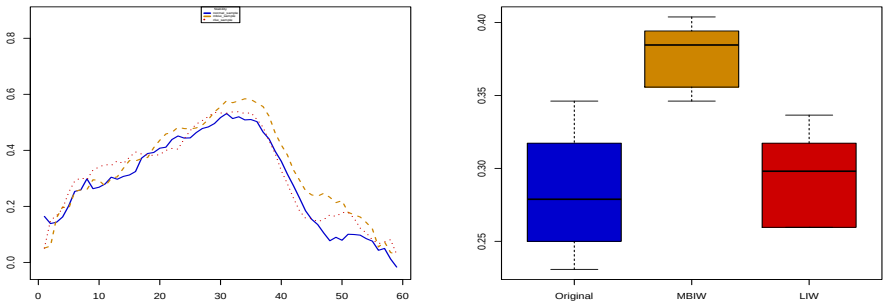



**(c)** mammogram stability **(d)** mammogram error




**(e)** musk stability **(f)** musk error

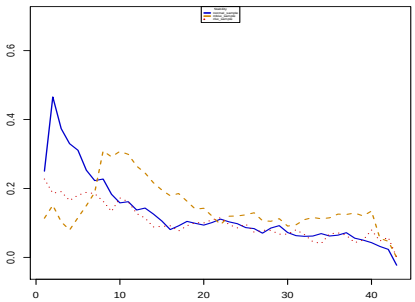
**Figure C.10:** Feature stability on UCI data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



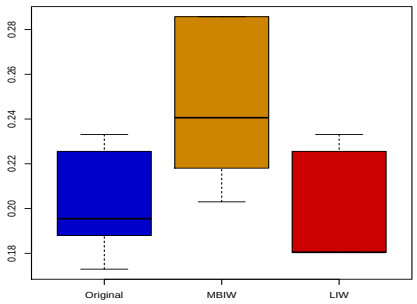

**(a)** sonar stability



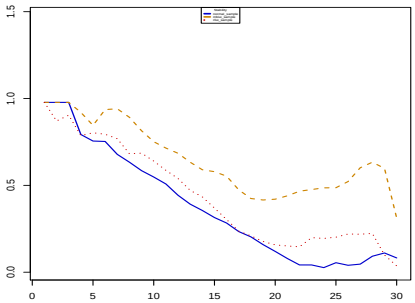

**(b)** sonar error



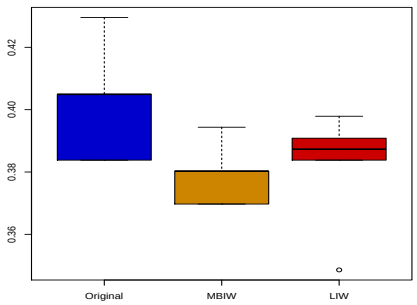

**(c)** spectf stability




**(d)** spectf error

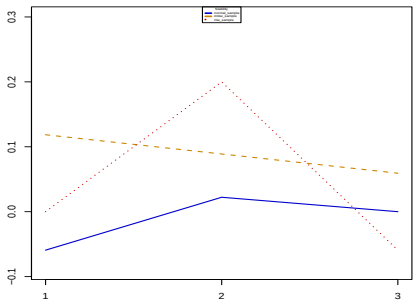



**(e)** wdbc stability

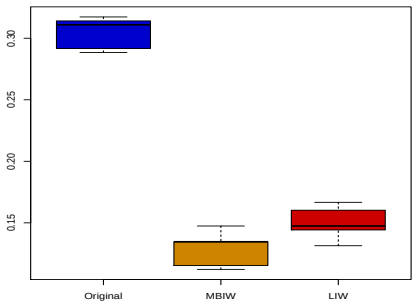



**(f)** wdbc error

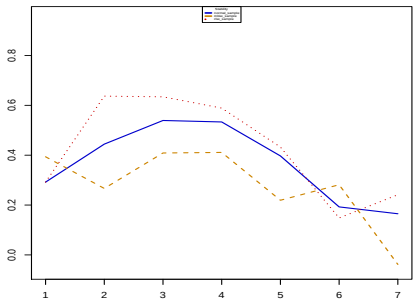
**Figure C.11:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



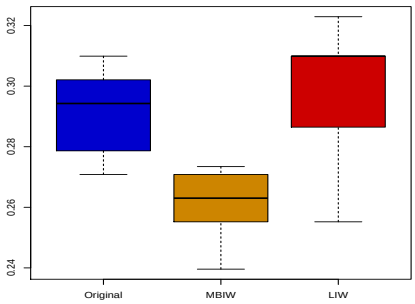
**(a)** balance scale stability



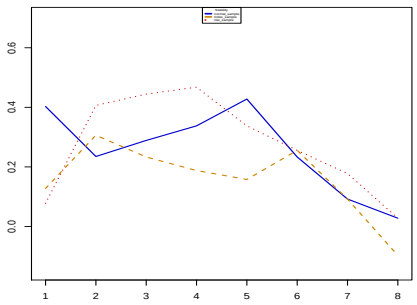

**(b)** balance scale error



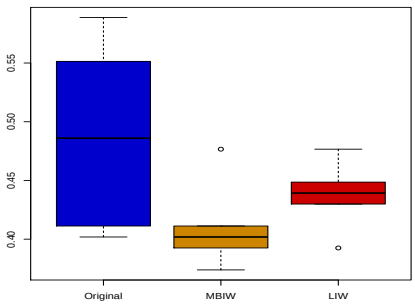
**(c)** diabetes stability




**(d)** diabetes error

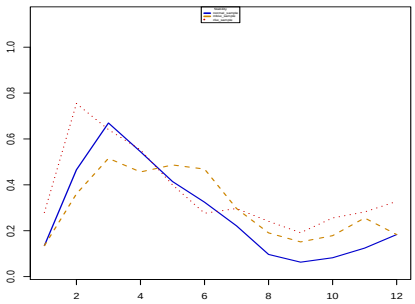


**(e)** glass stability

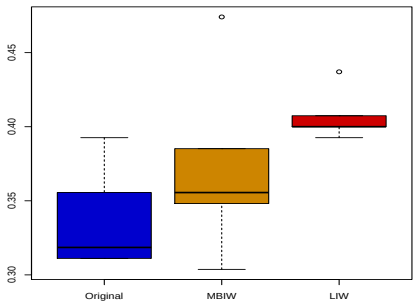



**(f)** glass error

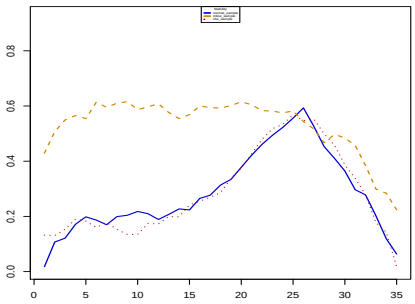
**Figure C.12:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



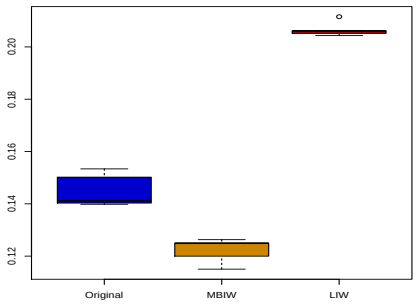
**(a)** heart statlog stability



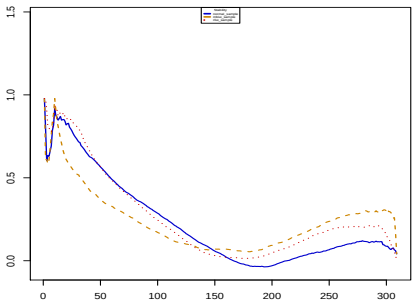

**(b)** heart statlog error



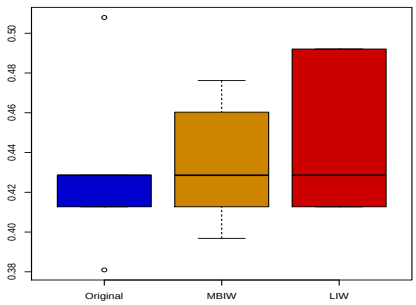
**(c)** landsat stability




**(d)** landsat error

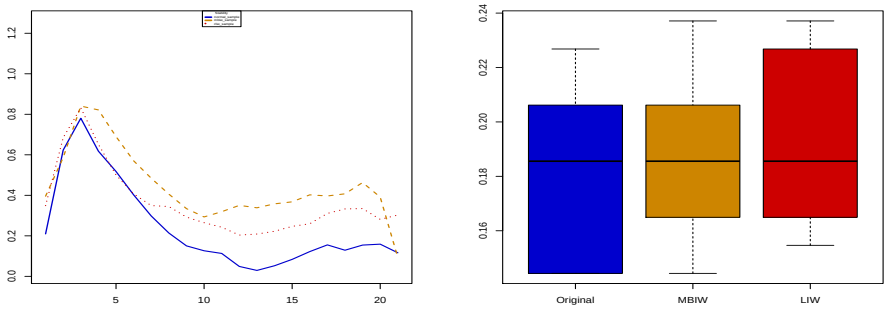


**(e)** lsvt voice stability




**(f)** lsvt voice error

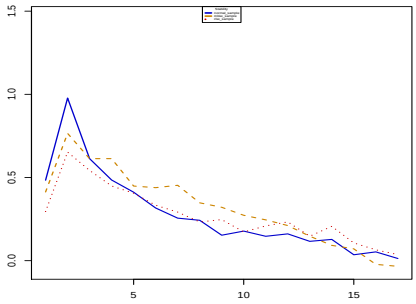
**Figure C.13:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



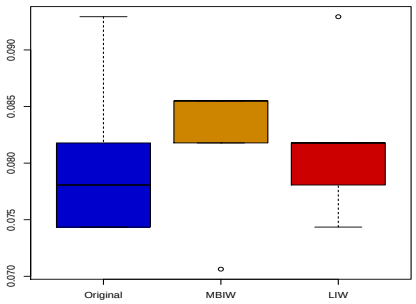
**(a)** parkinsons statlog stability



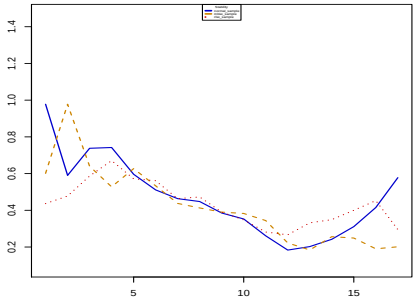

**(b)** parkinsons statlog error



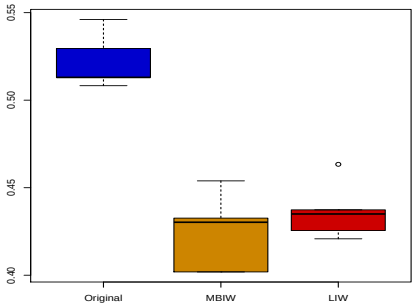
**(c)** pop failures stability




**(d)** pop failures error

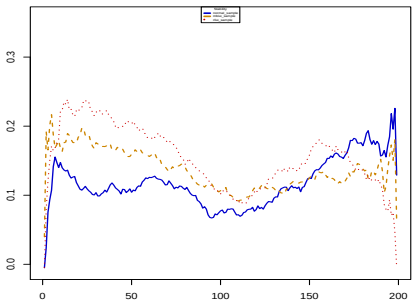


**(e)** vehicle stability

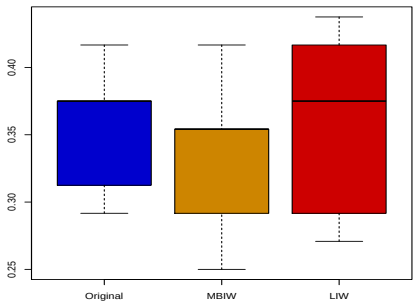



**(f)** vechicle voice error

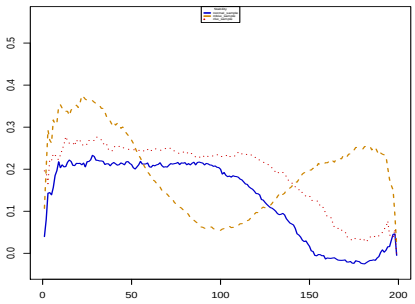
**Figure C.14:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



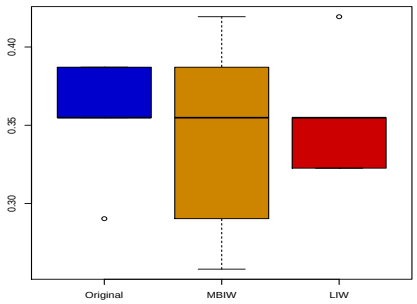

**(a)** breast cancer stability



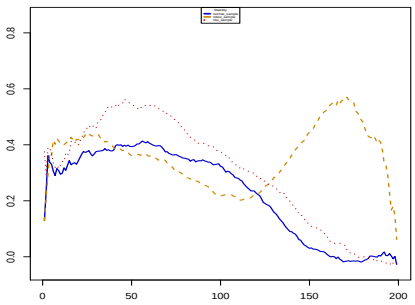

**(b)** breast cancer error



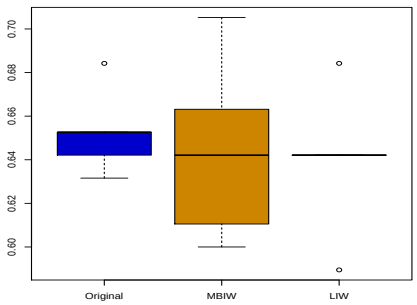

**(c)** colon tumor stability




**(d)** colon tumor error

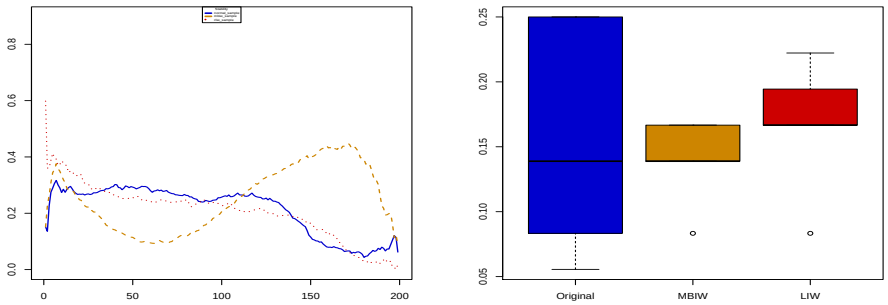



**(e)** gcm stability

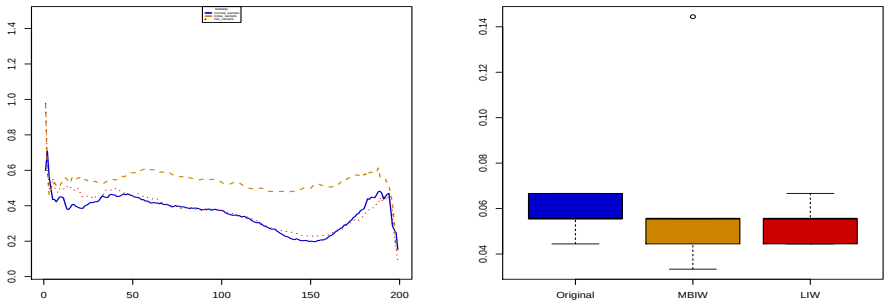



**(f)** gcm error

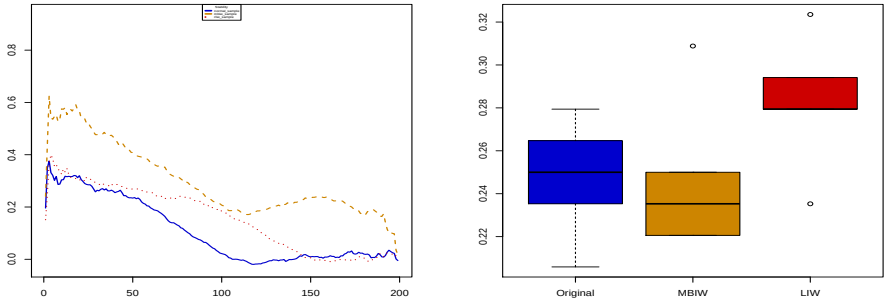
**Figure C.15:** Feature stability on microarray data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** version). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** leukemia stability **(b)** leukemia error

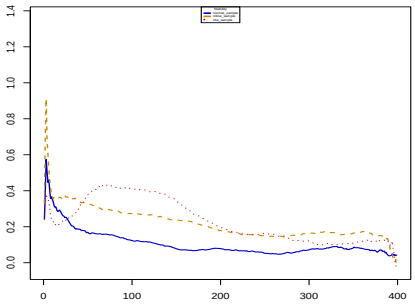



**(c)** lung cancer stability **(d)** lung cancer error

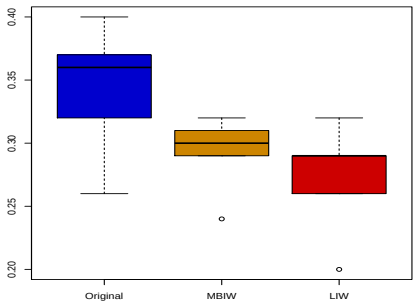



**(e)** prostate cancer stability **(f)** prostate cancer error

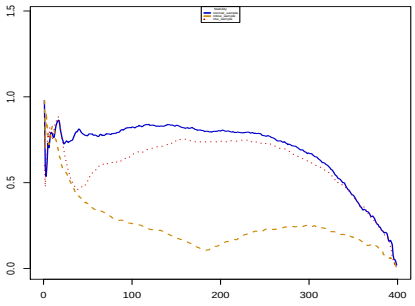
**Figure C.16:** Feature stability on microarray data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



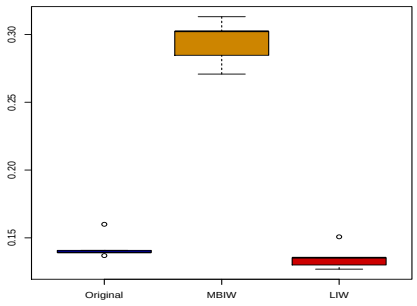

**(a)** arcene stability



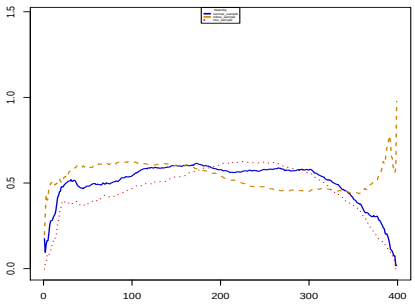

**(b)** arcene error



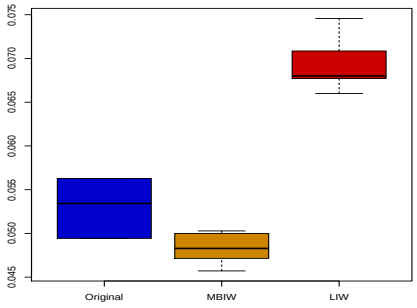

**(c)** madelon stability




**(d)** madelon error

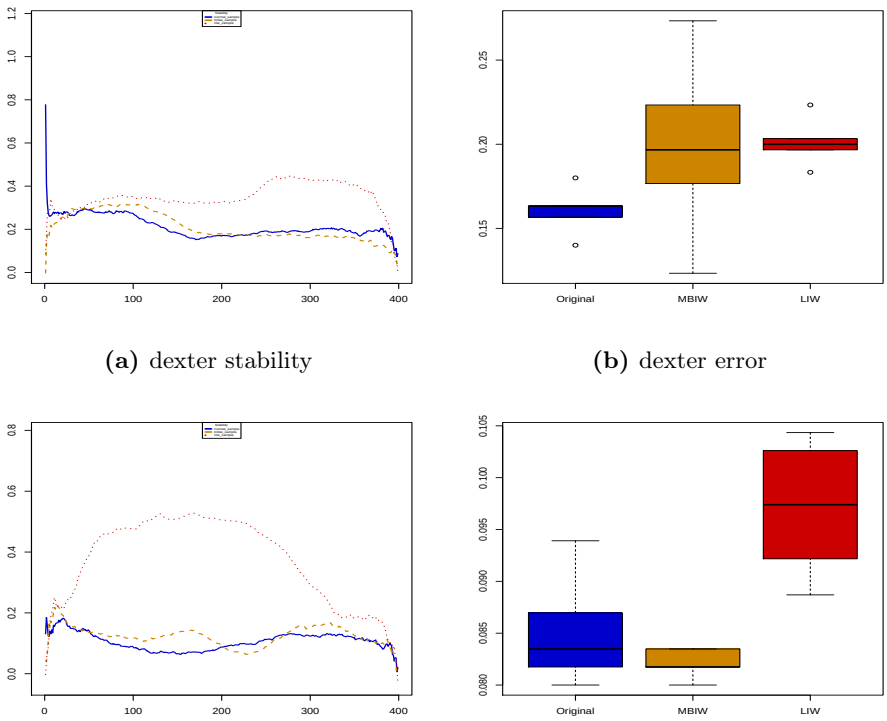



**(e)** gisette stability




**(f)** gisette error

**Figure C.17:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



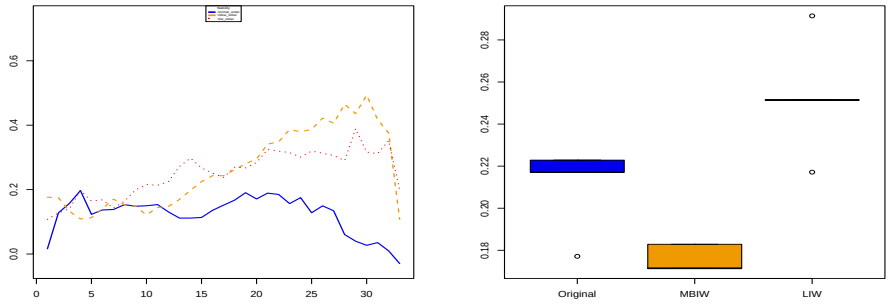

**(a)** dexter stability

**(b)** dexter error

**(c)** dorothea stability

**(d)** dorothea error

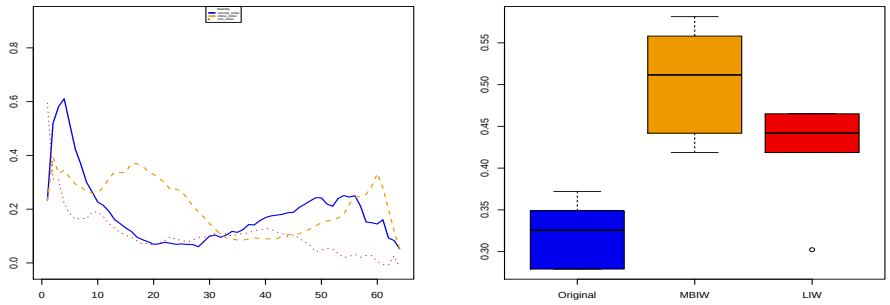
**Figure C.18:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** ionosphere stability



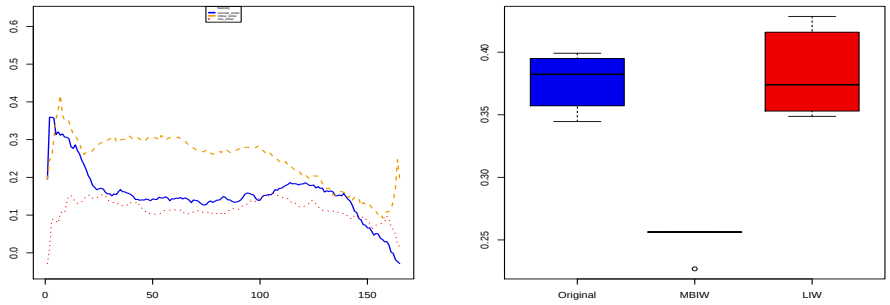

**(b)** ionosphere error




**(c)** mammogram stability




**(d)** mammogram error

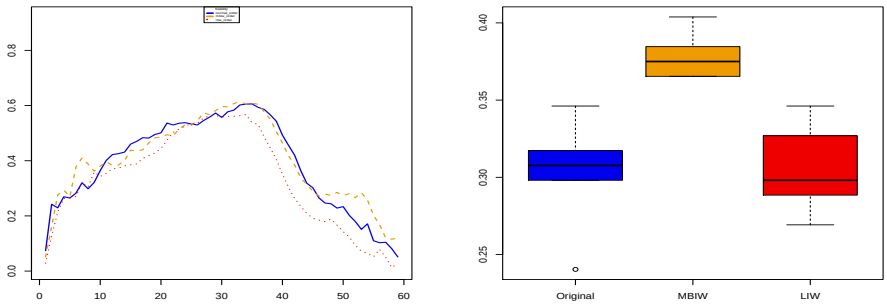



**(e)** musk stability




**(f)** musk error

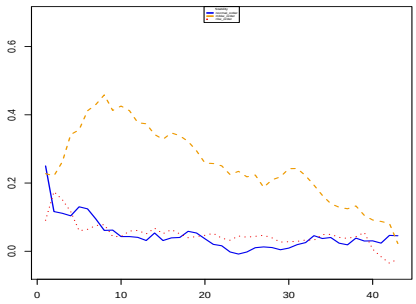
**Figure C.19:** Feature stability on UCI data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



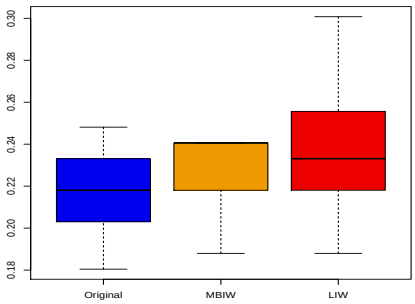

**(a)** sonar stability



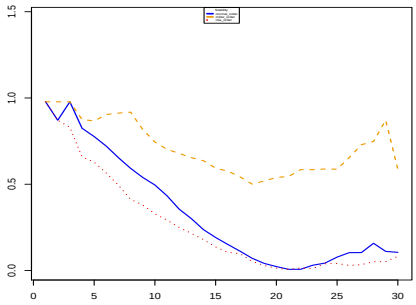

**(b)** sonar error



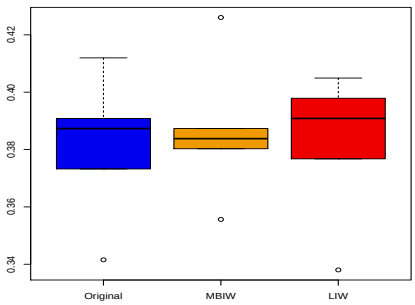

**(c)** spectf stability




**(d)** spectf error

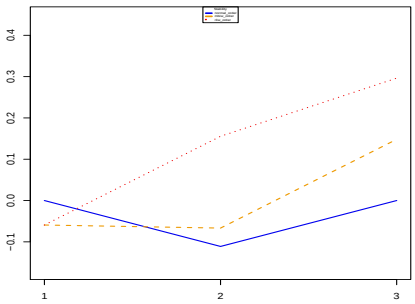



**(e)** wdbc stability

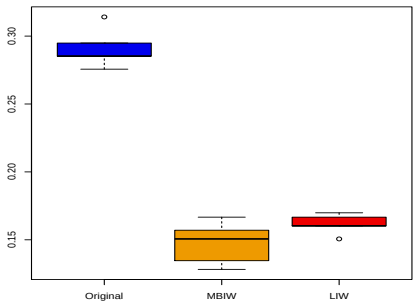



**(f)** wdbc error

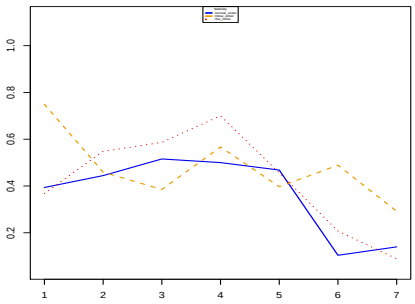
**Figure C.20:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



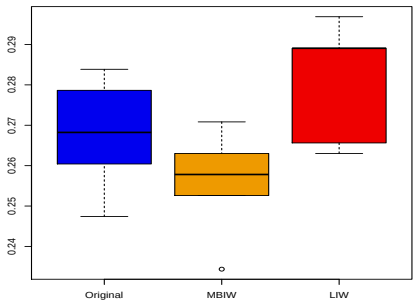
**(a)** balance scale stability



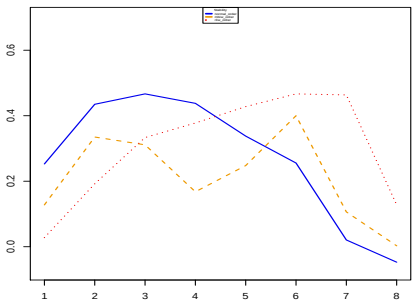

**(b)** balance scale error



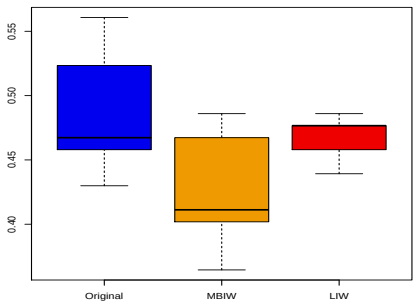
**(c)** diabetes stability




**(d)** diabetes error

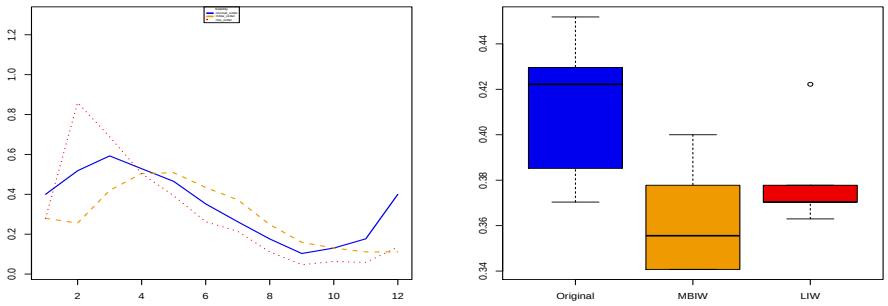


**(e)** glass stability

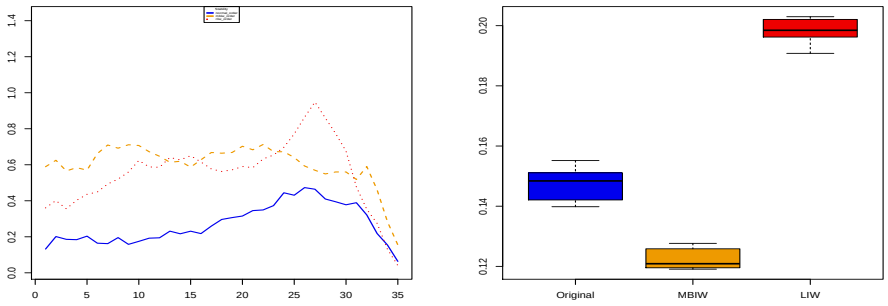



**(f)** glass error

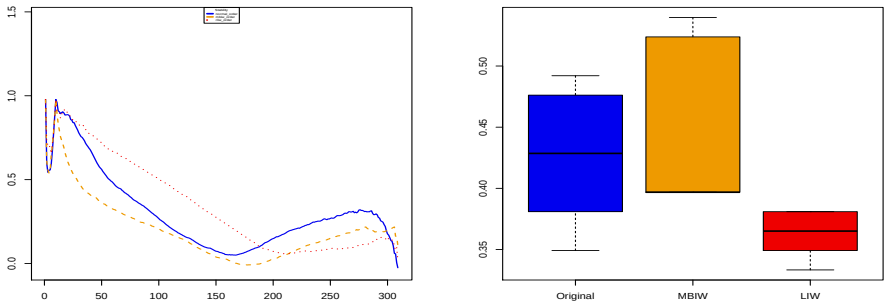
**Figure C.21:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** heart statlog stability **(b)** heart statlog error

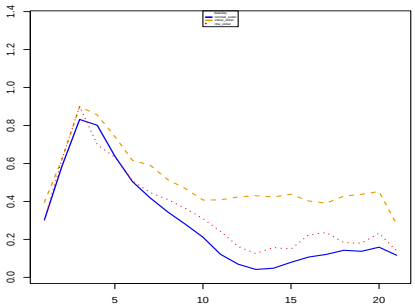



**(c)** landsat stability **(d)** landsat error

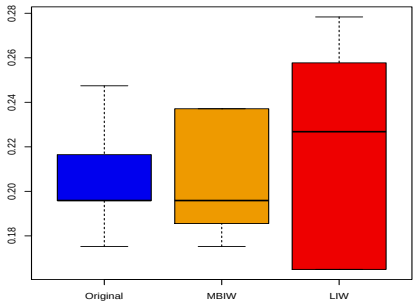



**(e)** lsvt voice stability **(f)** lsvt voice error

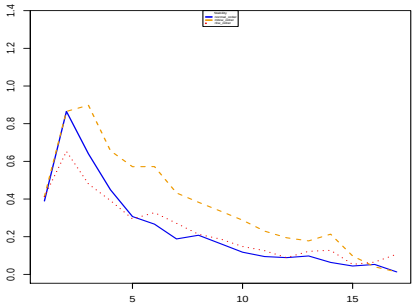
**Figure C.22:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



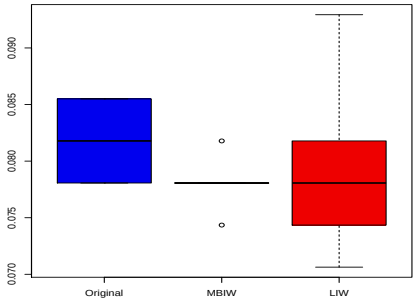

**(a)** parkinsons statlog stability



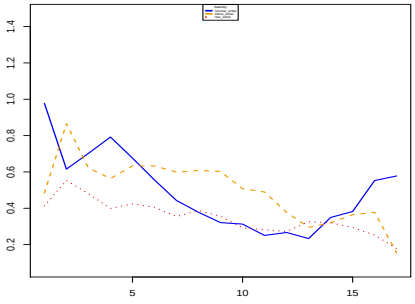

**(b)** parkinsons statlog error



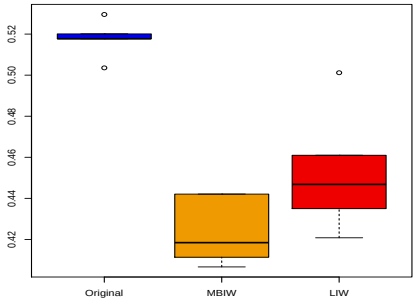

**(c)** pop failures stability




**(d)** pop failures error

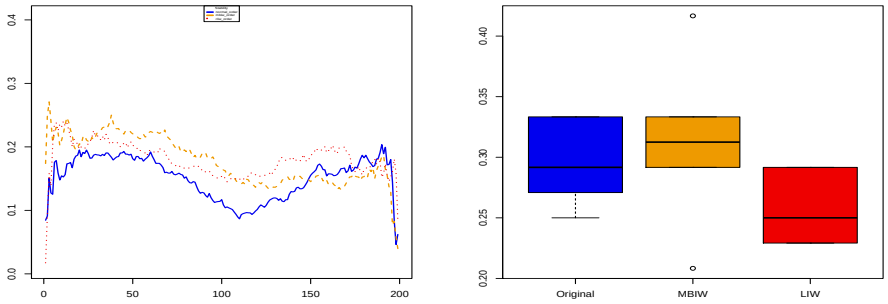



**(e)** vehicle stability




**(f)** vechicle voice error

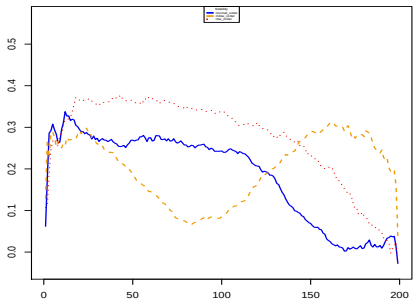
**Figure C.23:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



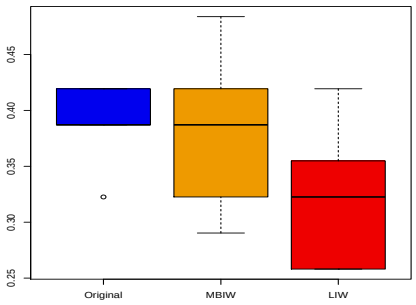

**(a)** breast cancer stability



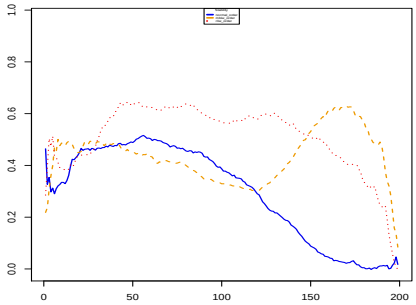

**(b)** breast cancer error



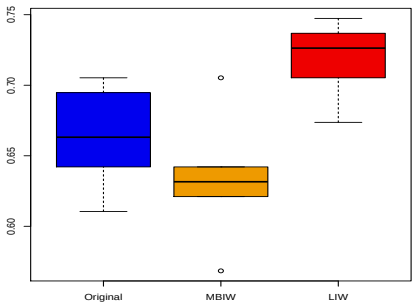

**(c)** colon tumor stability




**(d)** colon tumor error

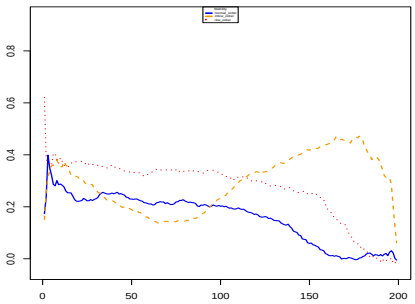



**(e)** gcm stability

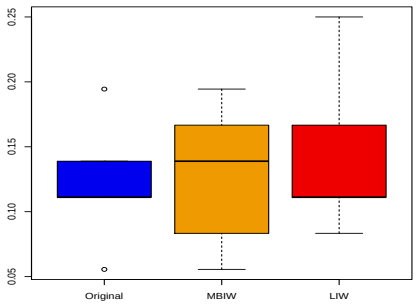



**(f)** gcm error

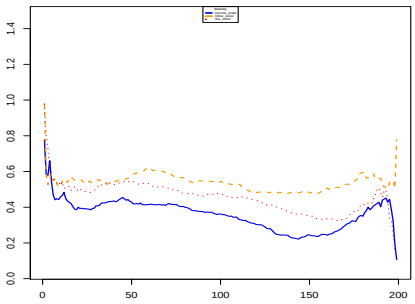
**Figure C.24:** Feature stability on microarray data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.



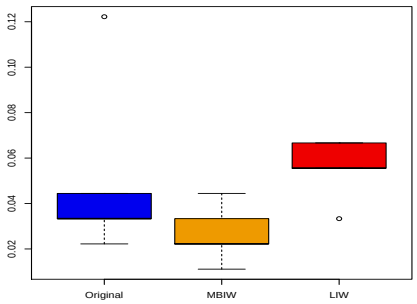
**(a)** leukemia stability



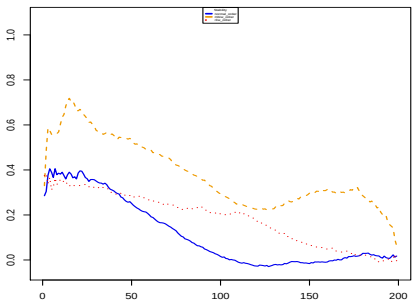

**(b)** leukemia error



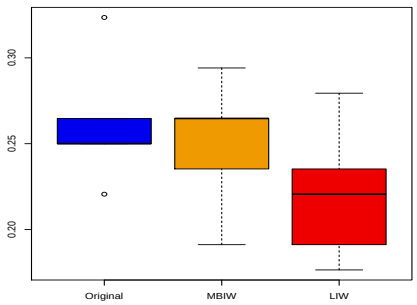
**(c)** lung cancer stability




**(d)** lung cancer error

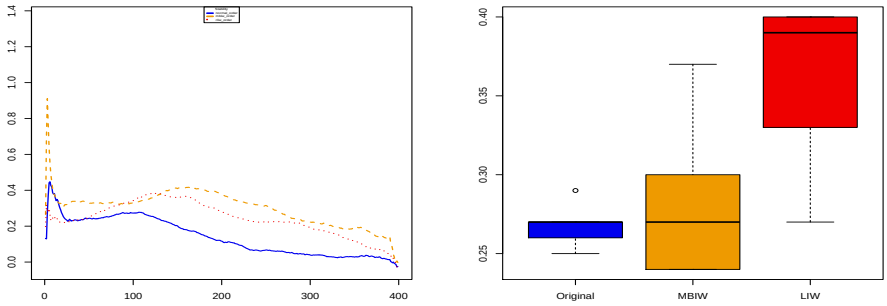


**(e)** prostate cancer stability

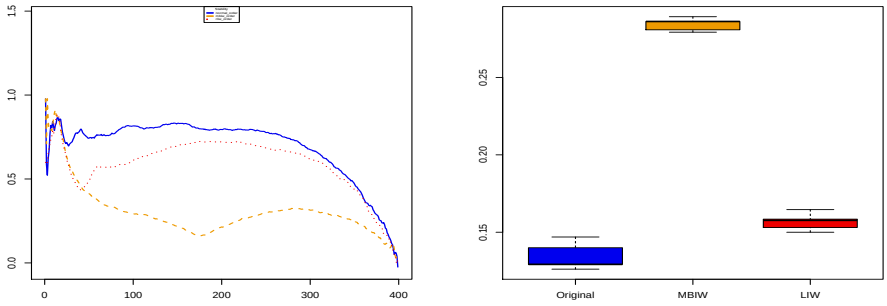



**(f)** prostate cancer error

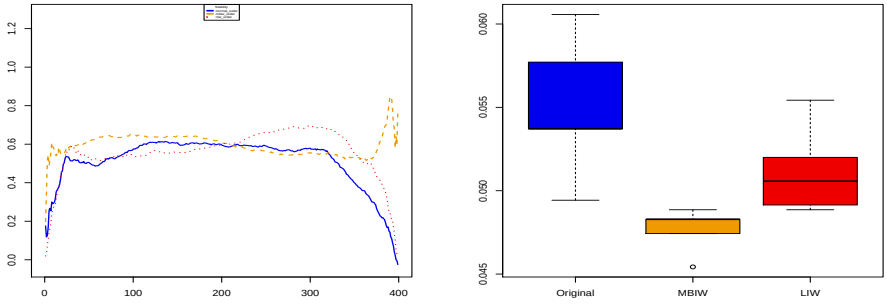
**Figure C.25:** Feature stability on microarray data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** arcene stability **(b)** arcene error

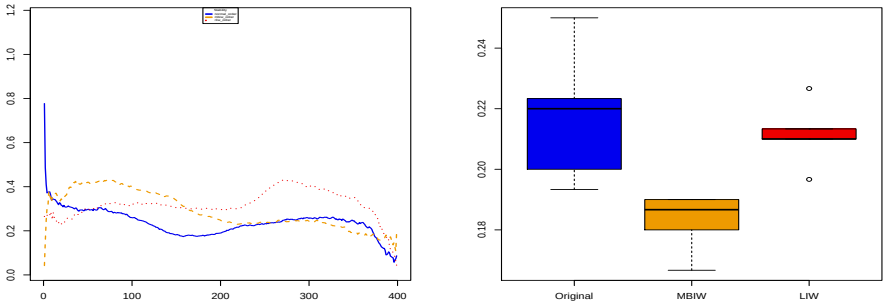



**(c)** madelon stability **(d)** madelon error

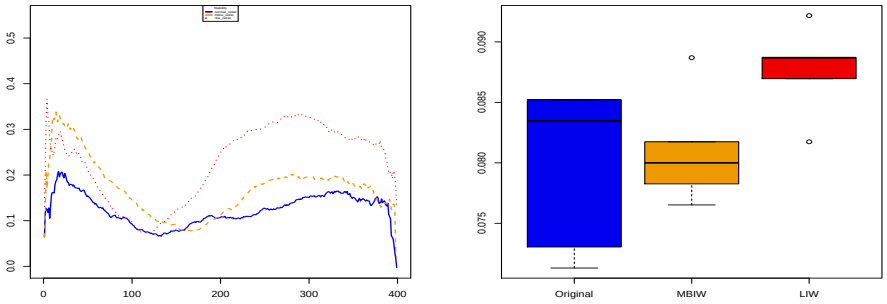



**(e)** gisette stability **(f)** gisette error

**Figure C.26:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.

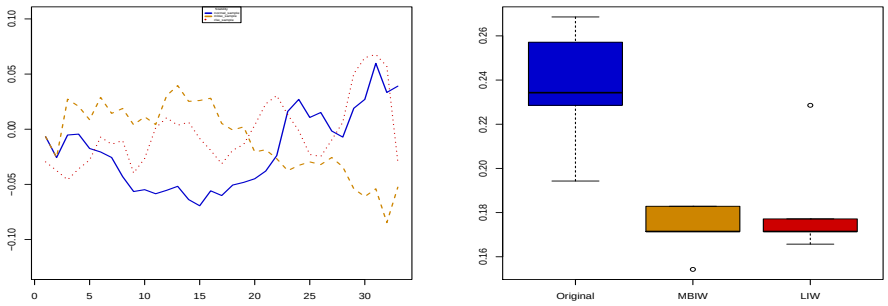



**(a)** dexter stability **(b)** dexter error

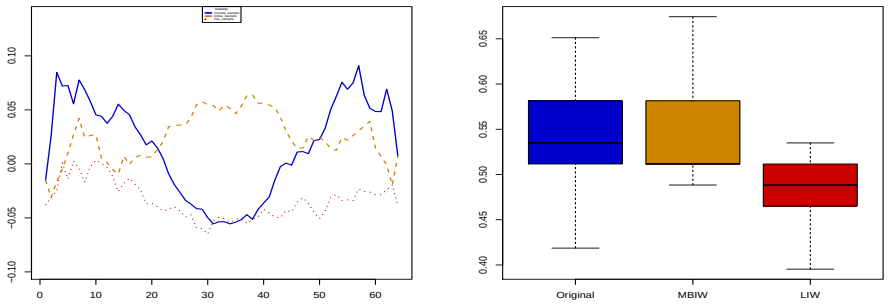



**(c)** dorothea stability **(d)** dorothea error

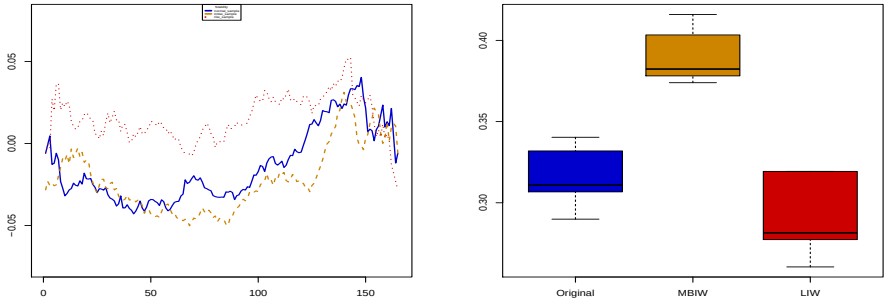
**Figure C.27:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Simba**, the dashed line is the weighted **SimbaMIW** version and the dotted one the weighted **SimbaLIW** verion). Right plot shows the average test errors for **Simba**, **SimbaMIW** and **SimbaLIW** respectively.




**(a)** ionosphere stability **(b)** ionosphere error

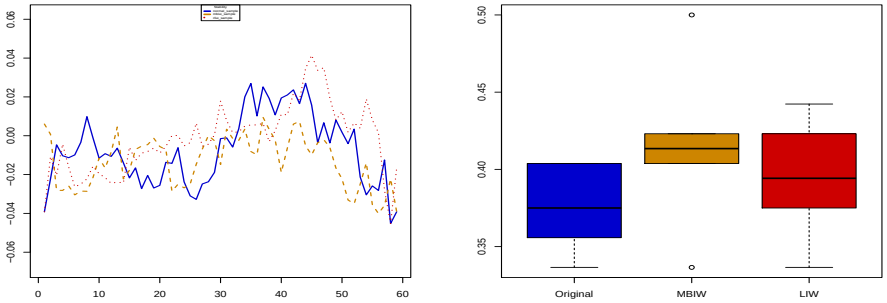



**(c)** mammogram stability **(d)** mammogram error

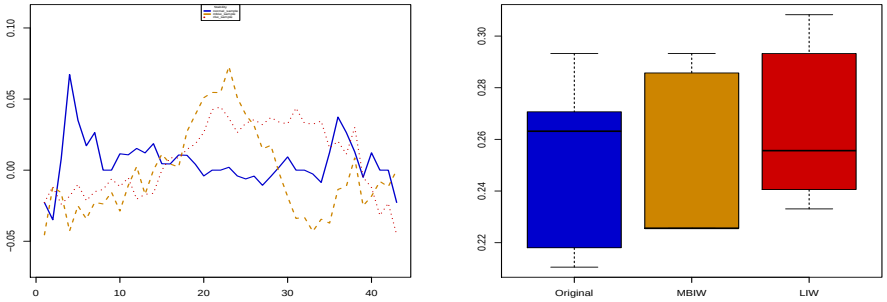



**(e)** musk stability **(f)** musk error

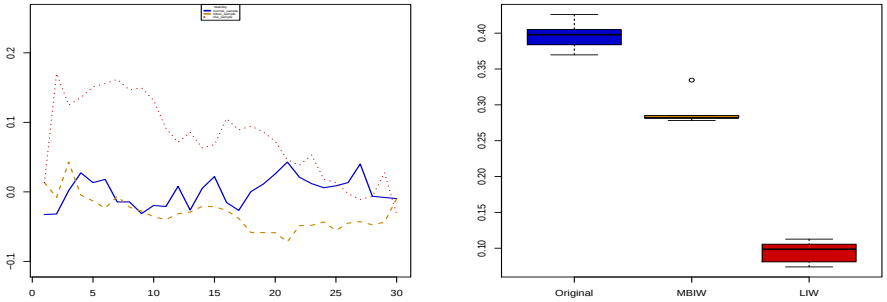
**Figure C.28:** Feature stability on UCI data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** sonar stability **(b)** sonar error

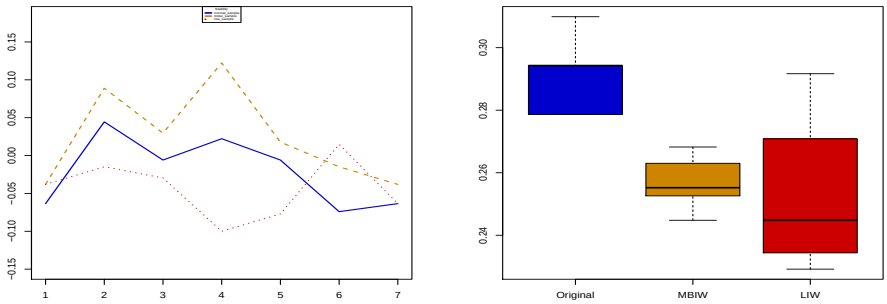



**(c)** spectf stability **(d)** spectf error



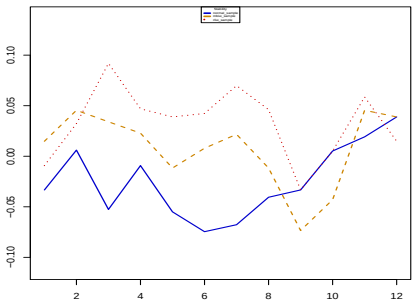

**(e)** wdbc stability **(f)** wdbc error

**Figure C.29:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.

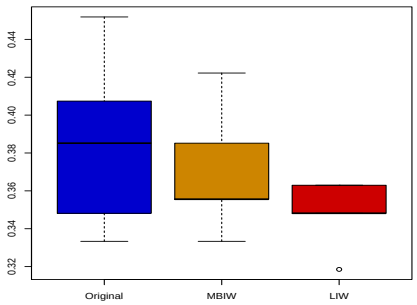



**(a)** diabetes stability **(b)** diabetes error

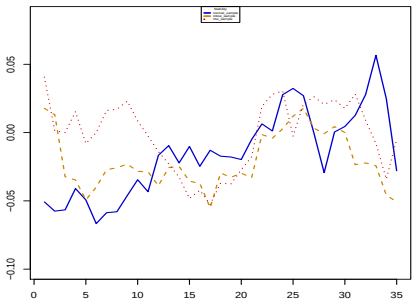
**Figure C.30:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



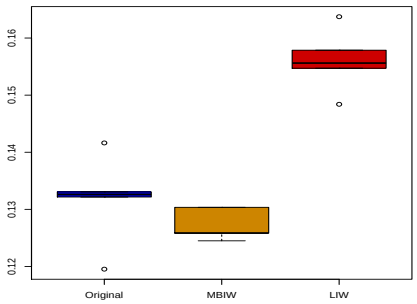
**(a)** heart statlog stability



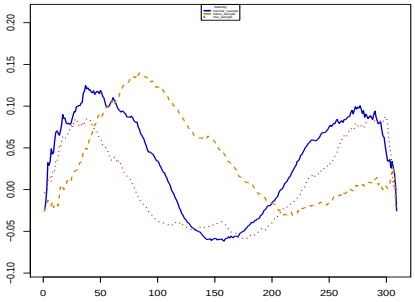

**(b)** heart statlog error



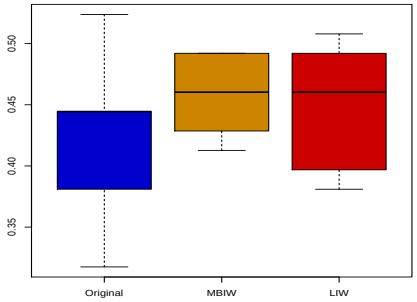
**(c)** landsat stability




**(d)** landsat error

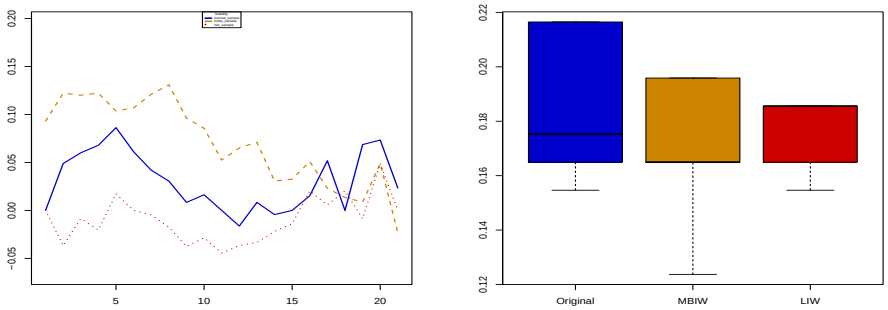


**(e)** lsvt voice stability

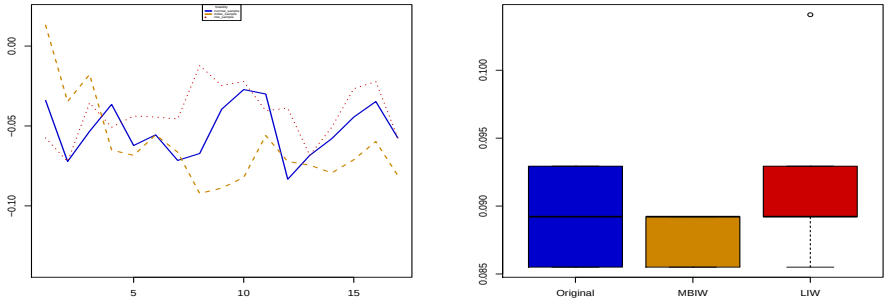



**(f)** lsvt voice error

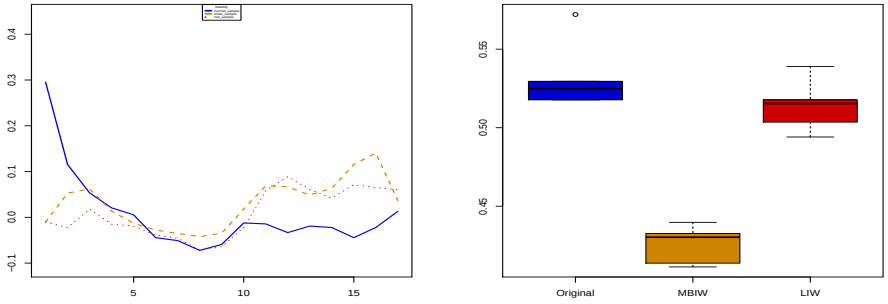
**Figure C.31:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** parkinsons statlog stability **(b)** parkinsons statlog error

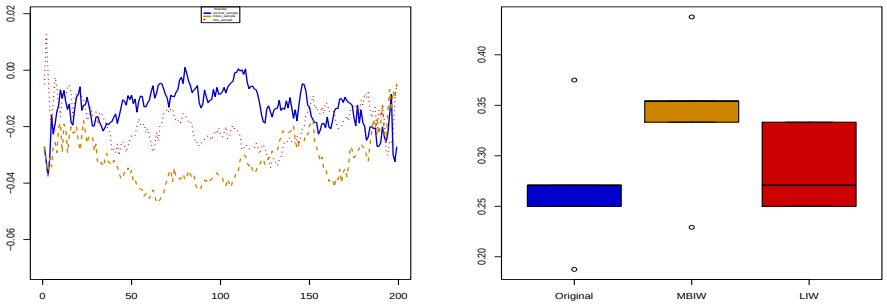



**(c)** pop failures stability **(d)** pop failures error




**(e)** vehicle stability **(f)** vechicle voice error

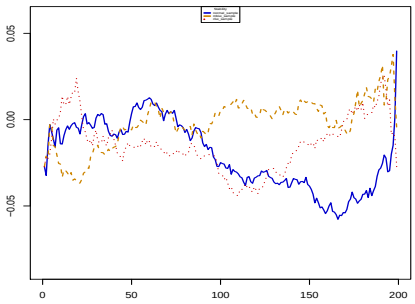
**Figure C.32:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



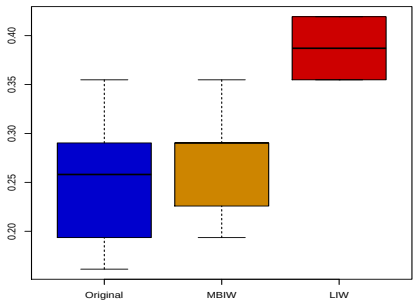

**(a)** breast cancer stability



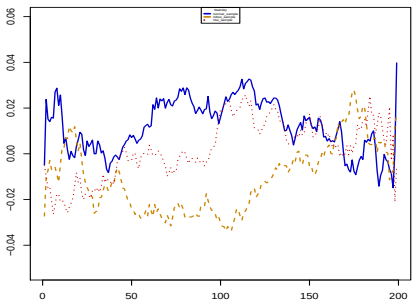

**(b)** breast cancer error



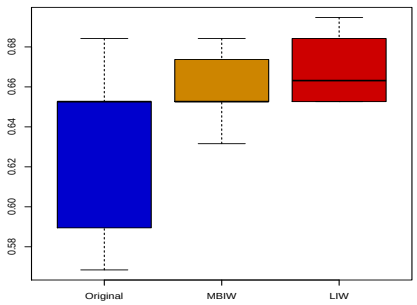

**(c)** colon tumor stability




**(d)** colon tumor error

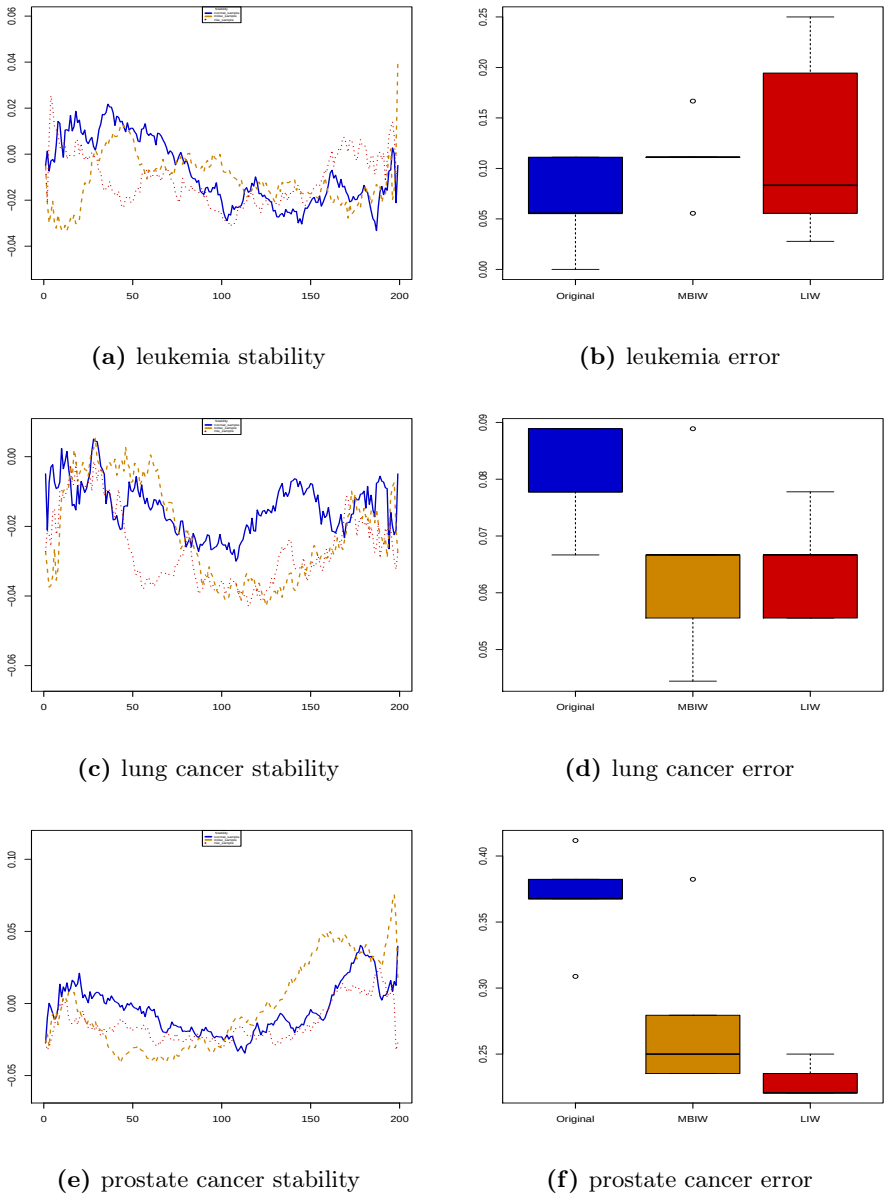



**(e)** gcm stability




**(f)** gcm error

**Figure C.33:** Feature stability on microarray data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** leukemia stability

**(b)** leukemia error

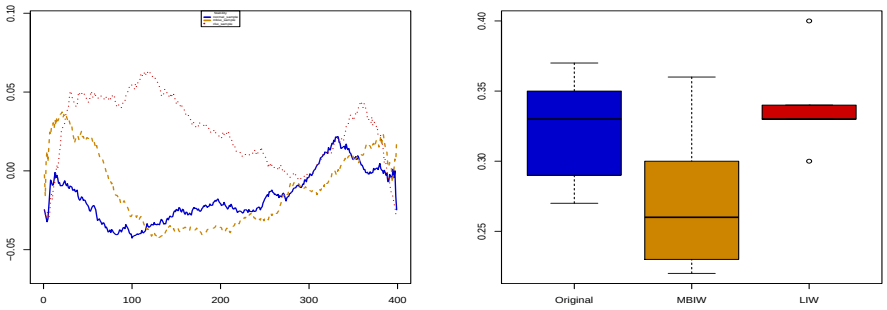
**(c)** lung cancer stability

**(d)** lung cancer error

**(e)** prostate cancer stability

**(f)** prostate cancer error

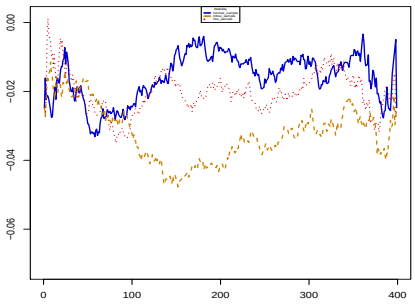
**Figure C.34:** Feature stability on microarray data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



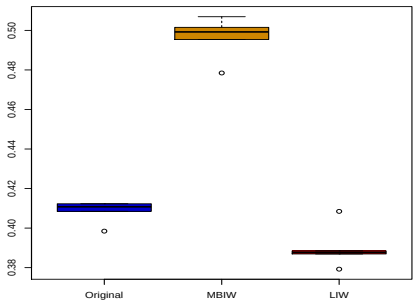

**(a)** arcene stability



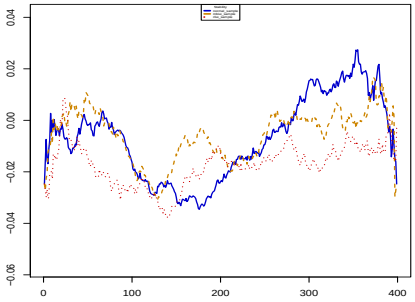

**(b)** arcene error



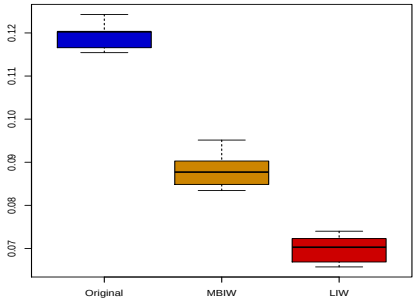

**(c)** madelon stability




**(d)** madelon error

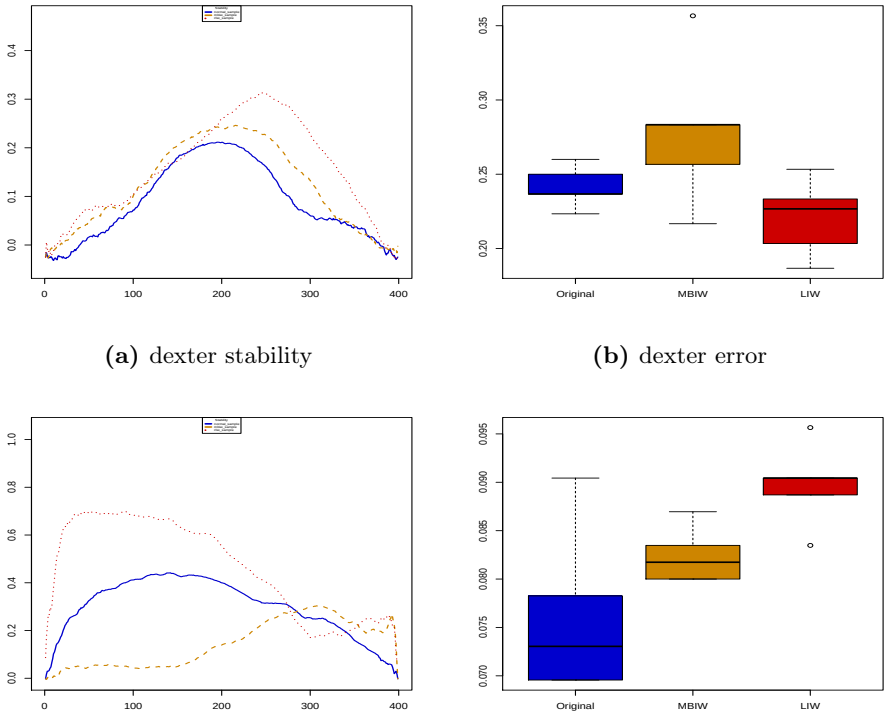



**(e)** gisette stability




**(f)** gisette error

**Figure C.35:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



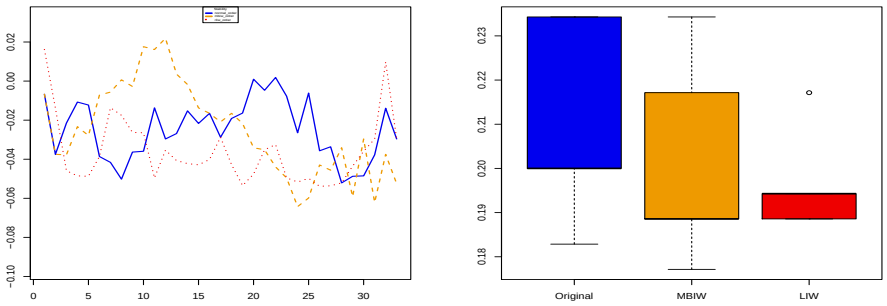

**(a)** dexter stability

**(b)** dexter error

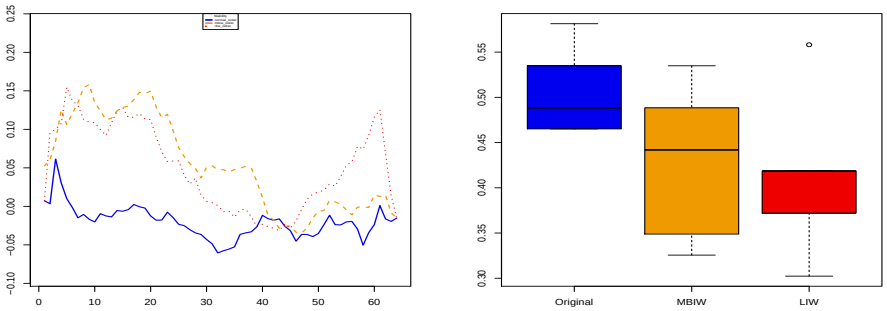
**(c)** dorothea stability

**(d)** dorothea error

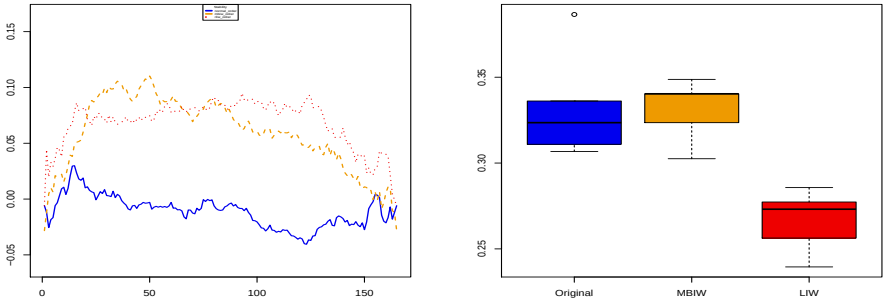
**Figure C.36:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** ionosphere stability **(b)** ionosphere error

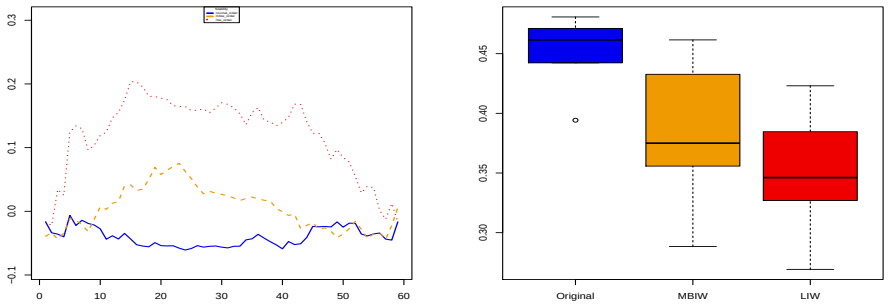



**(c)** mammogram stability **(d)** mammogram error

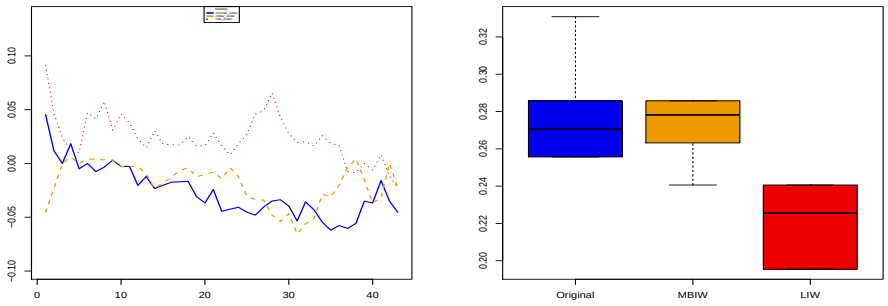



**(e)** musk stability **(f)** musk error

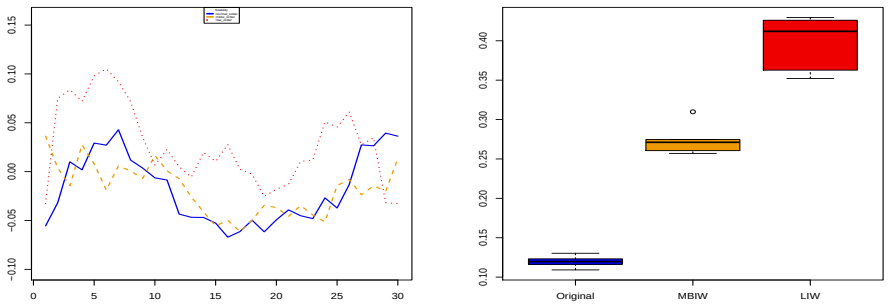
**Figure C.37:** Feature stability on UCI data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** sonar stability **(b)** sonar error

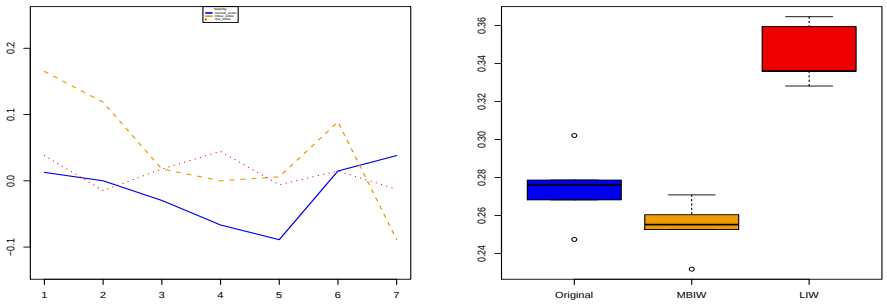



**(c)** spectf stability **(d)** spectf error



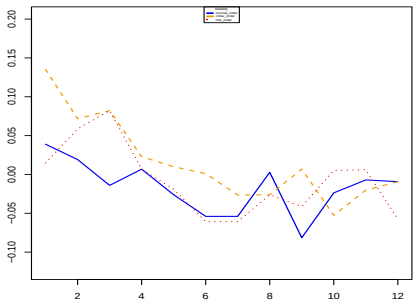

**(e)** wdbc stability **(f)** wdbc error

**Figure C.38:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.

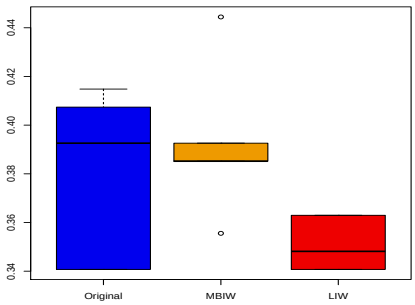



**(a)** diabetes stability **(b)** diabetes error

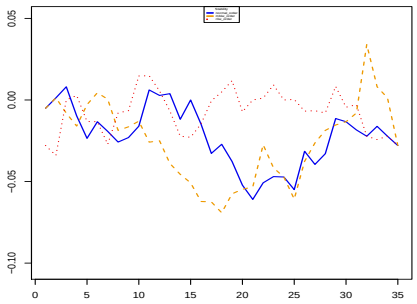
**Figure C.39:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



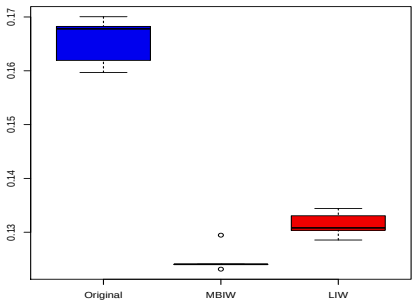

**(a)** heart statlog stability



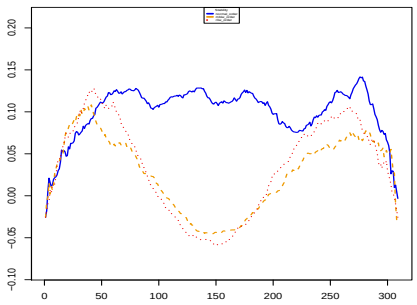

**(b)** heart statlog error



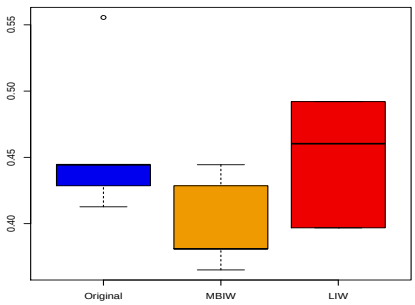

**(c)** landsat stability




**(d)** landsat error

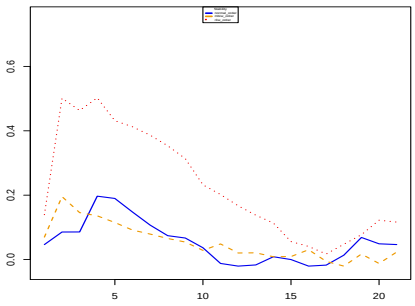



**(e)** lsvt voice stability

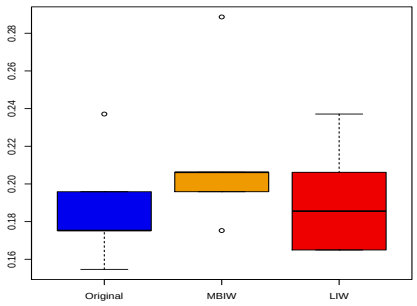



**(f)** lsvt voice error

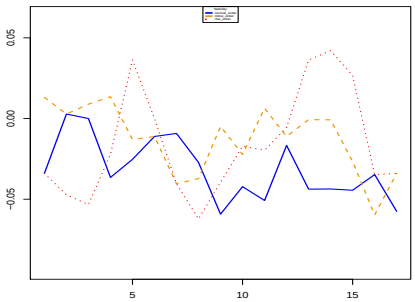
**Figure C.40:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



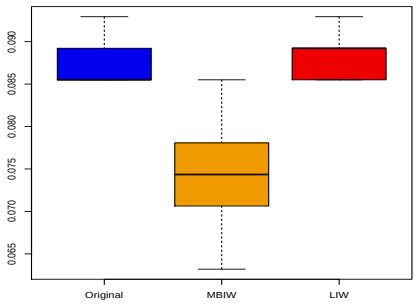

**(a)** parkinsons statlog stability



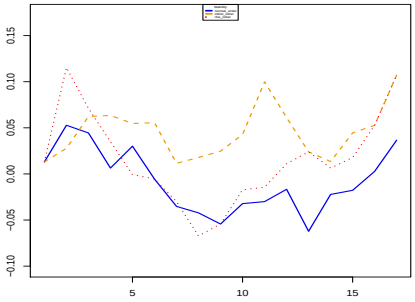

**(b)** parkinsons statlog error



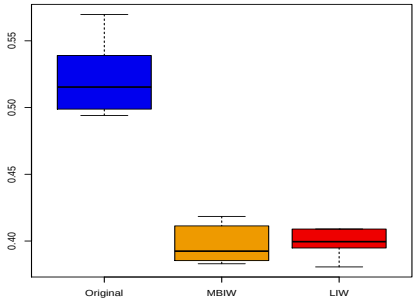

**(c)** pop failures stability




**(d)** pop failures error

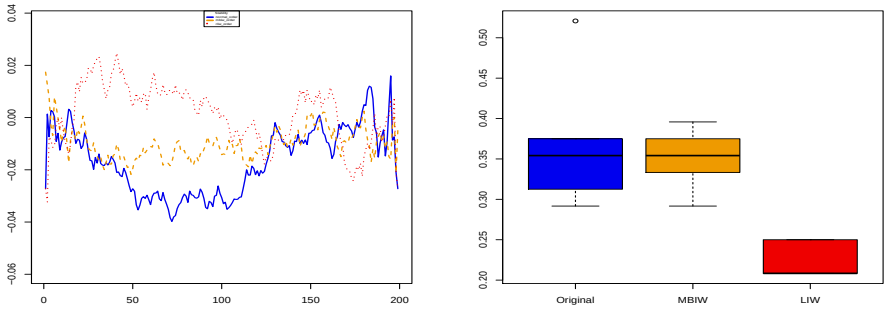



**(e)** vehicle stability




**(f)** vechicle voice error

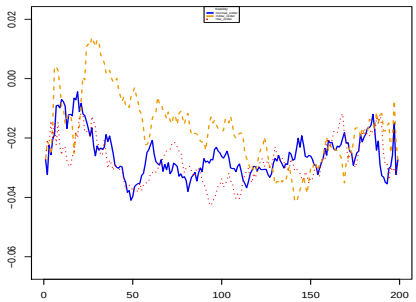
**Figure C.41:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



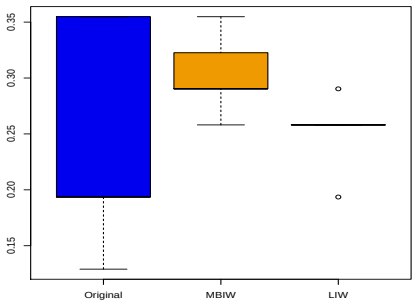

**(a)** breast cancer stability



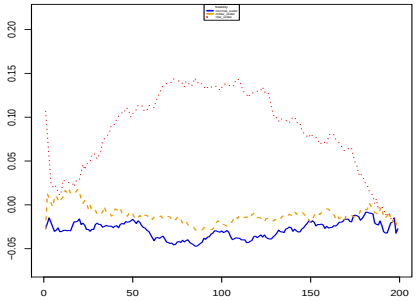

**(b)** breast cancer error



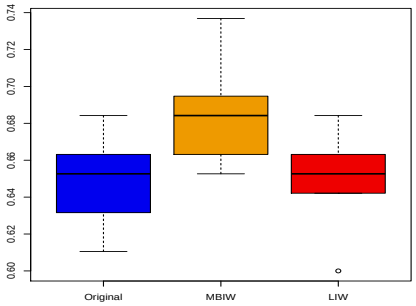

**(c)** colon tumor stability




**(d)** colon tumor error

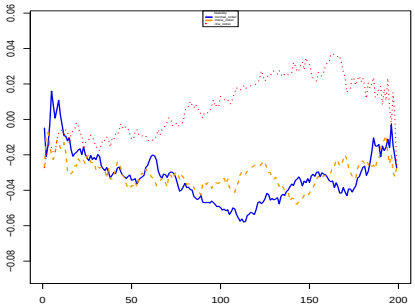



**(e)** gcm stability

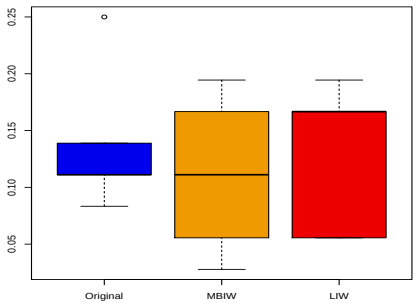



**(f)** gcm error

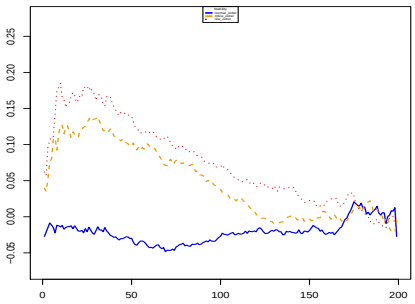
**Figure C.42:** Feature stability on microarray data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



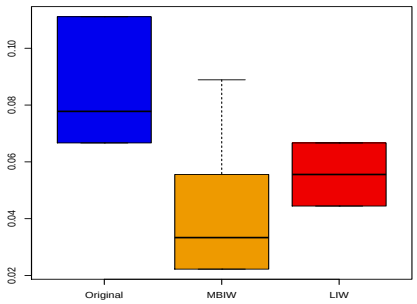

**(a)** leukemia stability



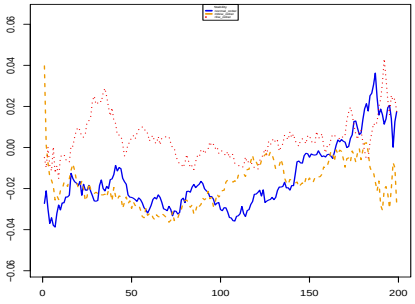

**(b)** leukemia error



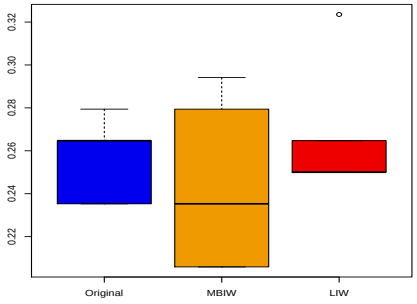

**(c)** lung cancer stability




**(d)** lung cancer error

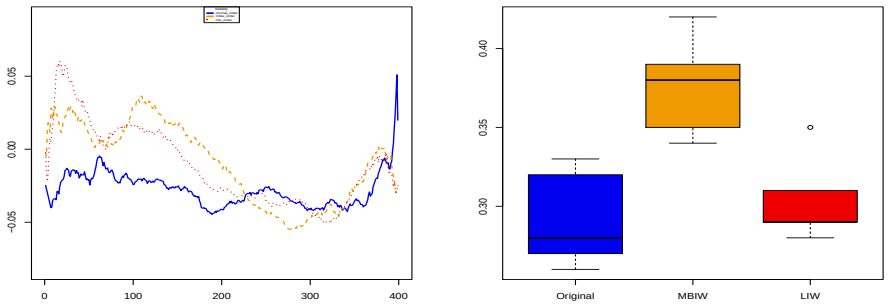



**(e)** prostate cancer stability




**(f)** prostate cancer error

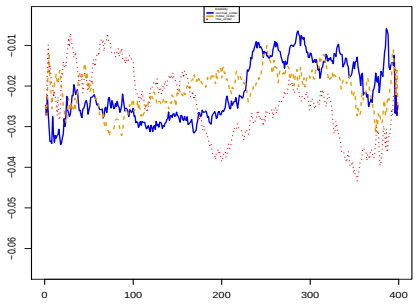
**Figure C.43:** Feature stability on microarray data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



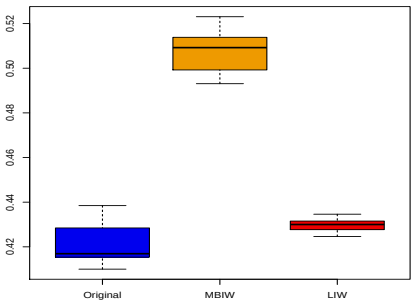

**(a)** arcene stability



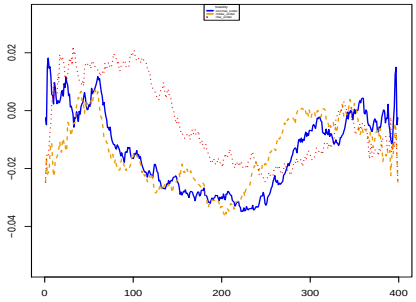

**(b)** arcene error



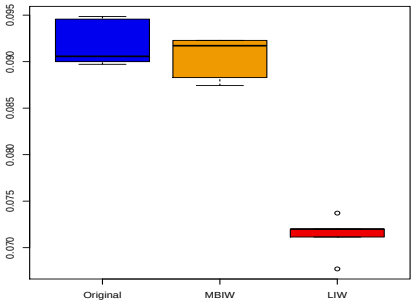

**(c)** madelon stability




**(d)** madelon error

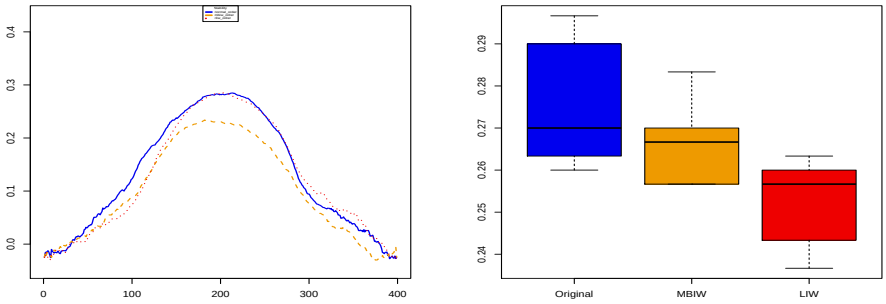



**(e)** gisette stability



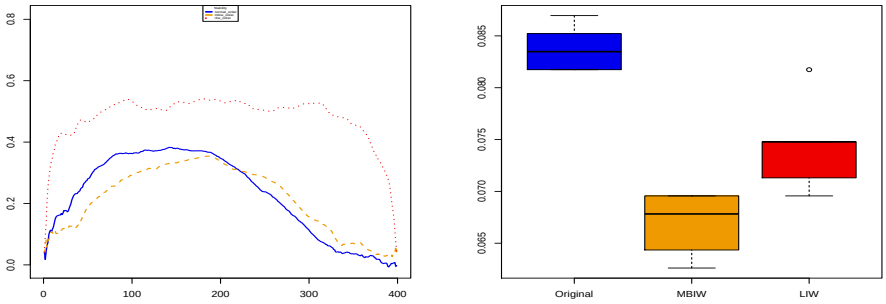

**(f)** gisette error

**Figure C.44:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.




**(a)** dexter stability

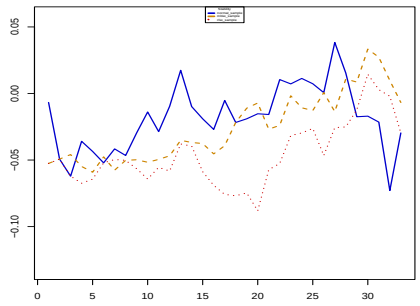
**(b)** dexter error



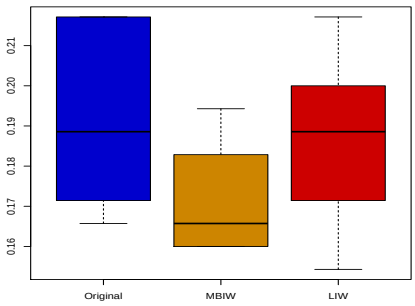

**(c)** dorothea stability

**(d)** dorothea error

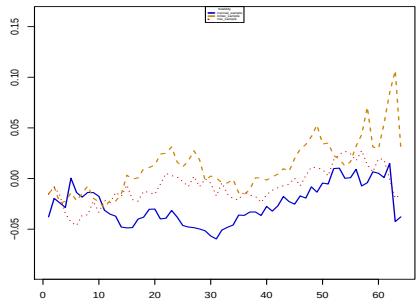
**Figure C.45:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **Relief**, the dashed line is the weighted **MBIW** + **Relief** version and the dotted one the weighted **RLIW** + **Relief** verion). Right plot shows the average test errors for **Relief**, **MBIW** + **Relief** and **RLIW** + **Relief** respectively.



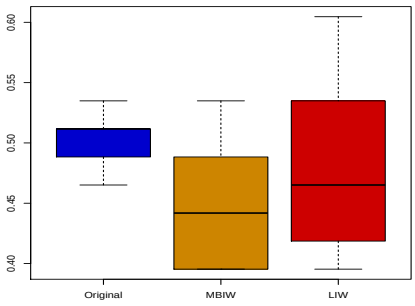
**(a)** ionosphere stability



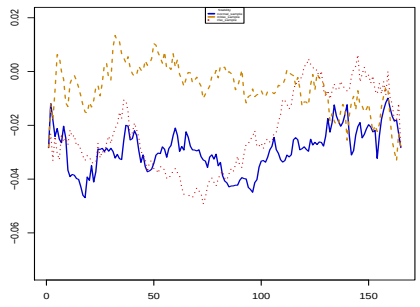

**(b)** ionosphere error



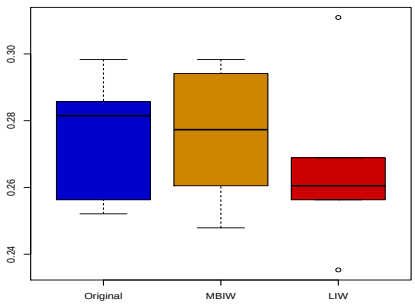
**(c)** mammogram stability




**(d)** mammogram error

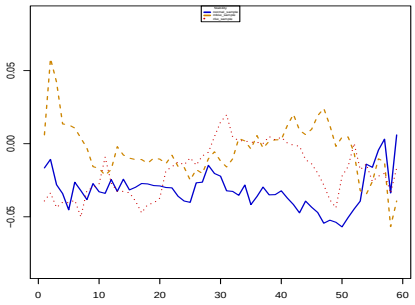


**(e)** musk stability

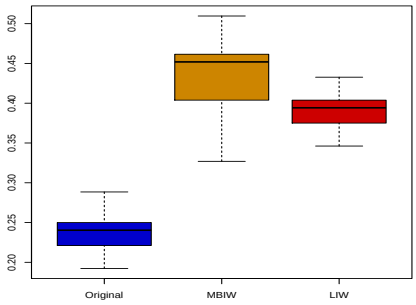



**(f)** musk error

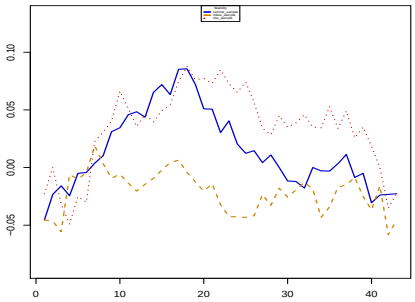
**Figure C.46:** Feature stability on UCI data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



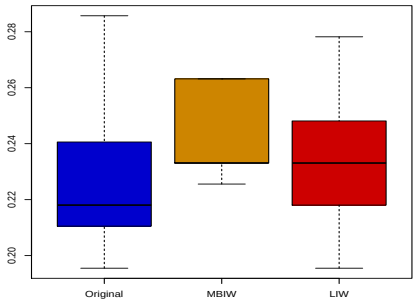
**(a)** sonar stability



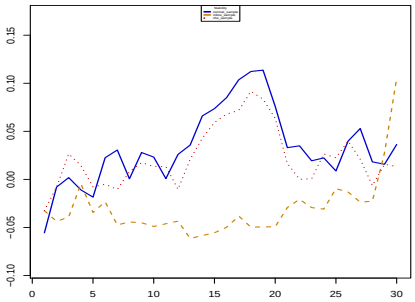

**(b)** sonar error



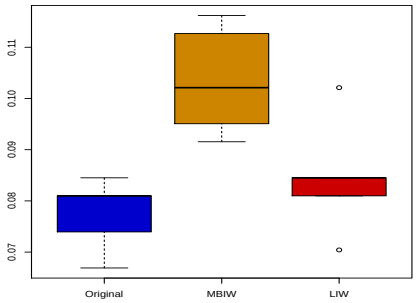
**(c)** spectf stability




**(d)** spectf error

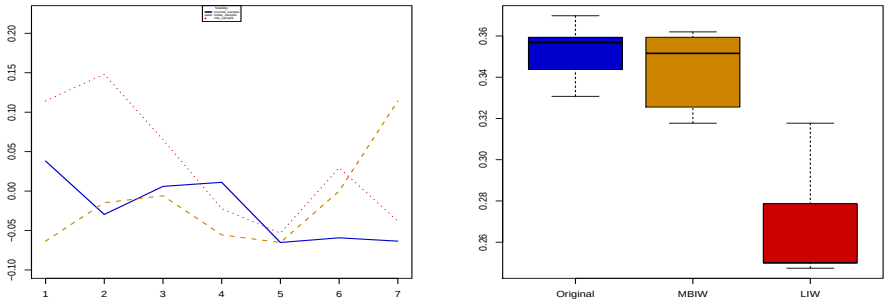


**(e)** wdbc stability



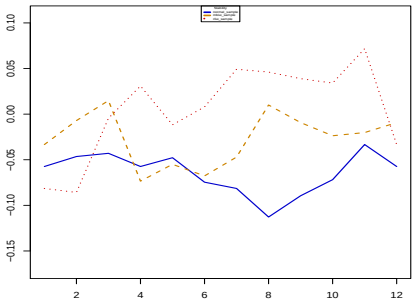

**(f)** wdbc error

**Figure C.47:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.

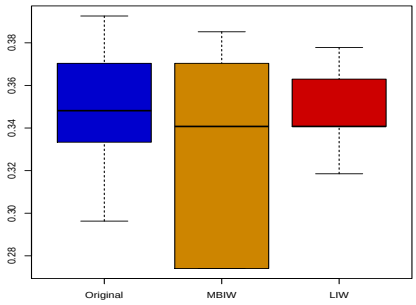



**(a)** diabetes stability **(b)** diabetes error

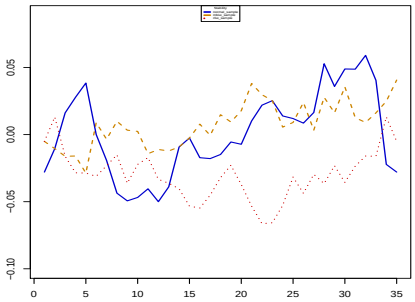
**Figure C.48:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



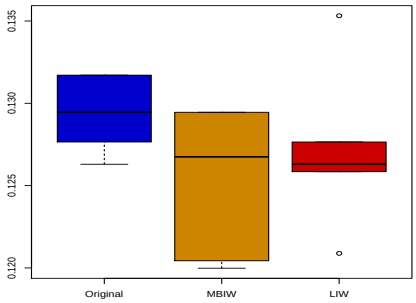
**(a)** heart statlog stability



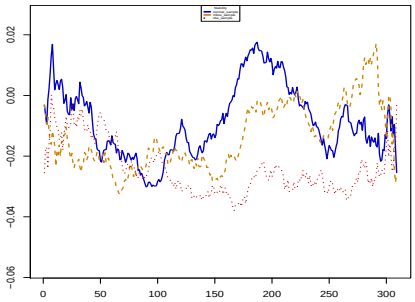

**(b)** heart statlog error



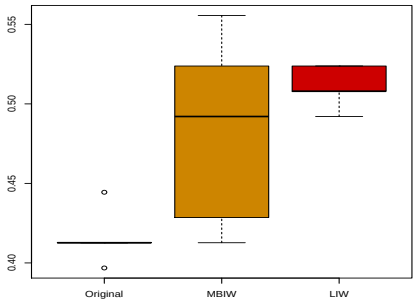
**(c)** landsat stability




**(d)** landsat error

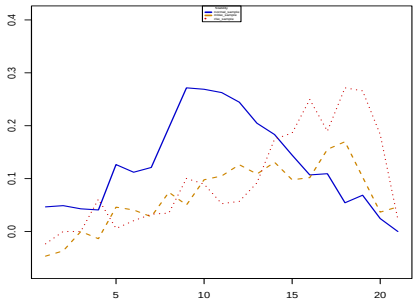


**(e)** lsvt voice stability

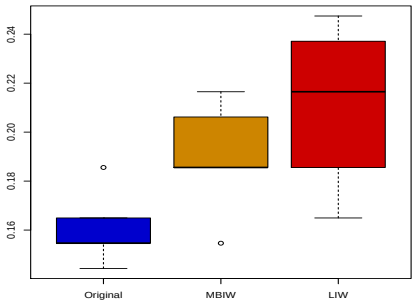



**(f)** lsvt voice error

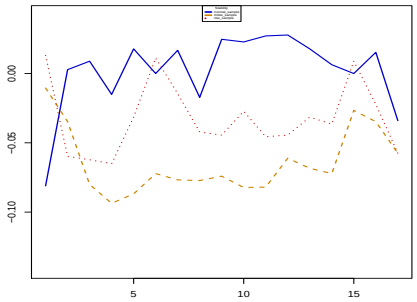
**Figure C.49:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



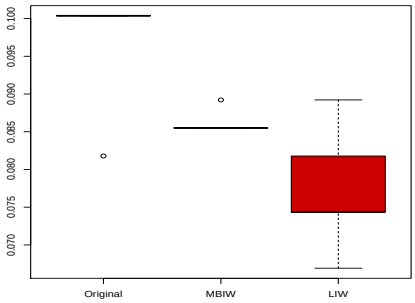

**(a)** parkinsons statlog stability



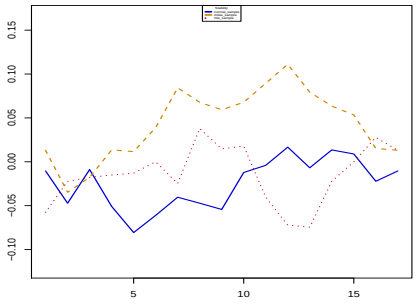

**(b)** parkinsons statlog error



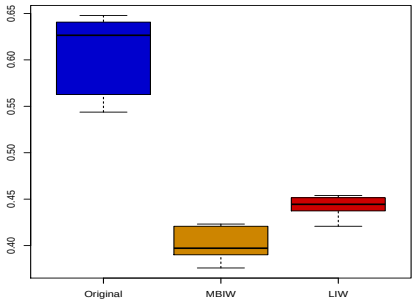

**(c)** pop failures stability




**(d)** pop failures error

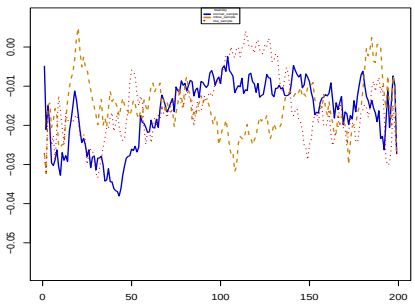



**(e)** vehicle stability

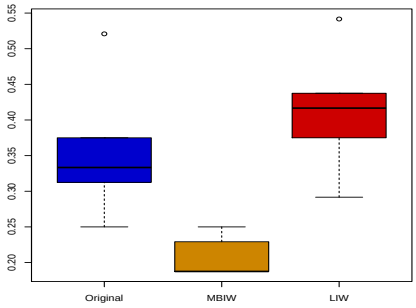



**(f)** vechicle voice error

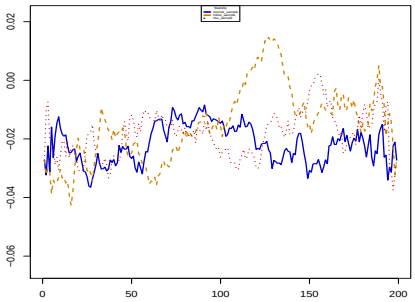
**Figure C.50:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



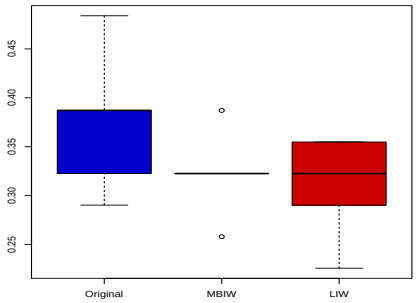

**(a)** breast cancer stability



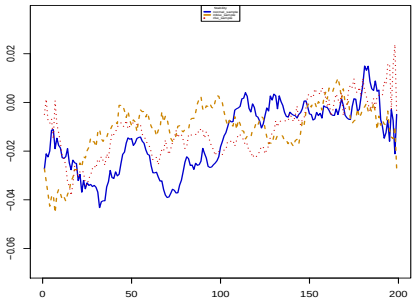

**(b)** breast cancer error



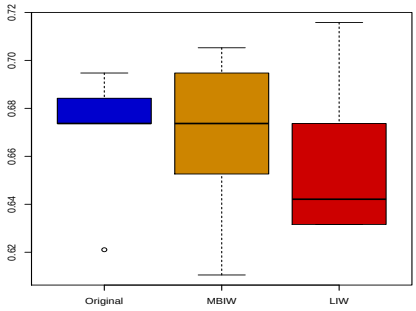

**(c)** colon tumor stability




**(d)** colon tumor error

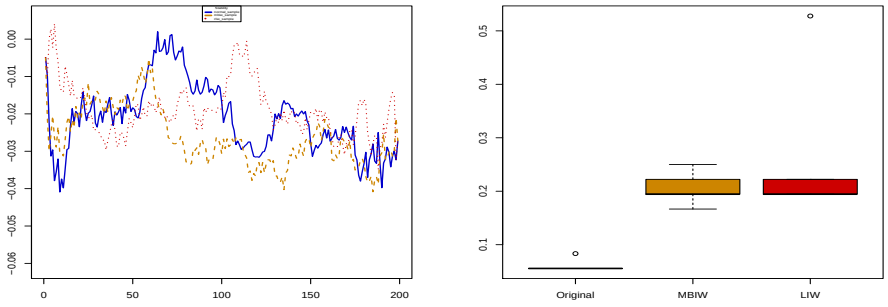



**(e)** gcm stability




**(f)** gcm error

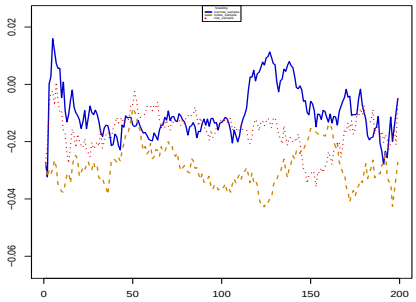
**Figure C.51:** Feature stability on microarray data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



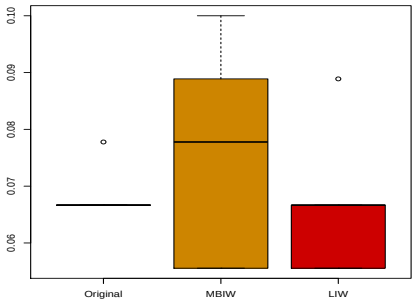

**(a)** leukemia stability



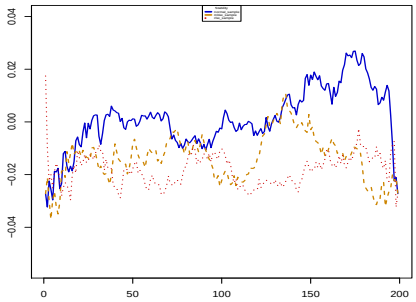

**(b)** leukemia error



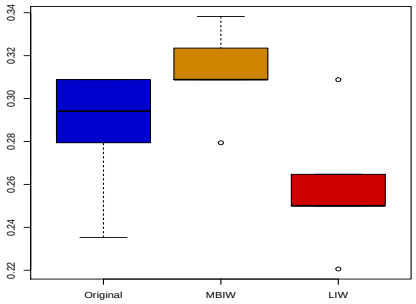

**(c)** lung cancer stability




**(d)** lung cancer error

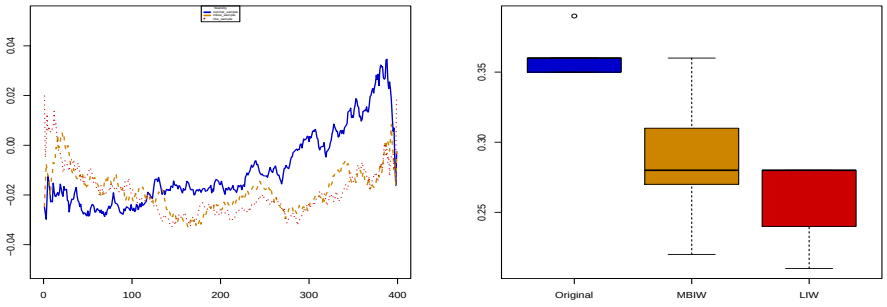



**(e)** prostate cancer stability

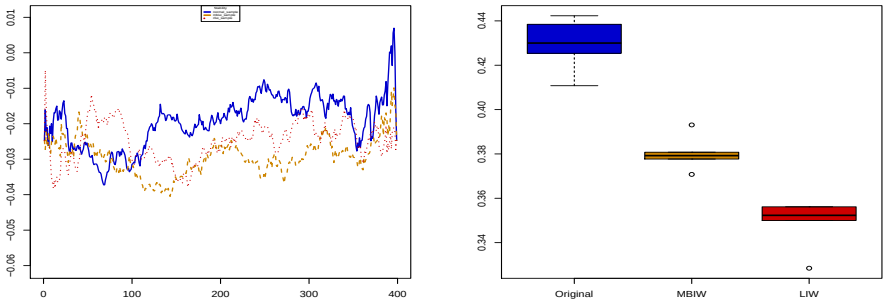



**(f)** prostate cancer error

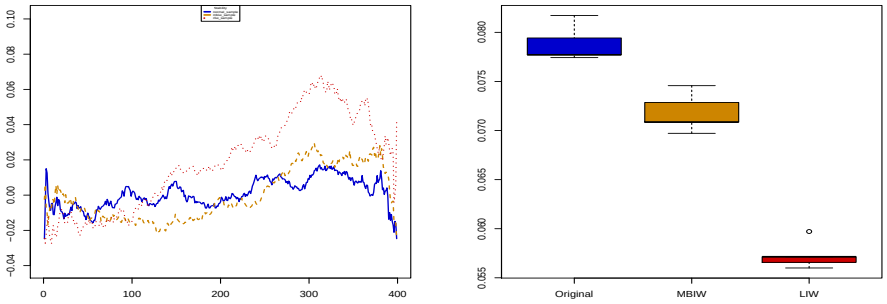
**Figure C.52:** Feature stability on microarray data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.




**(a)** arcene stability **(b)** arcene error

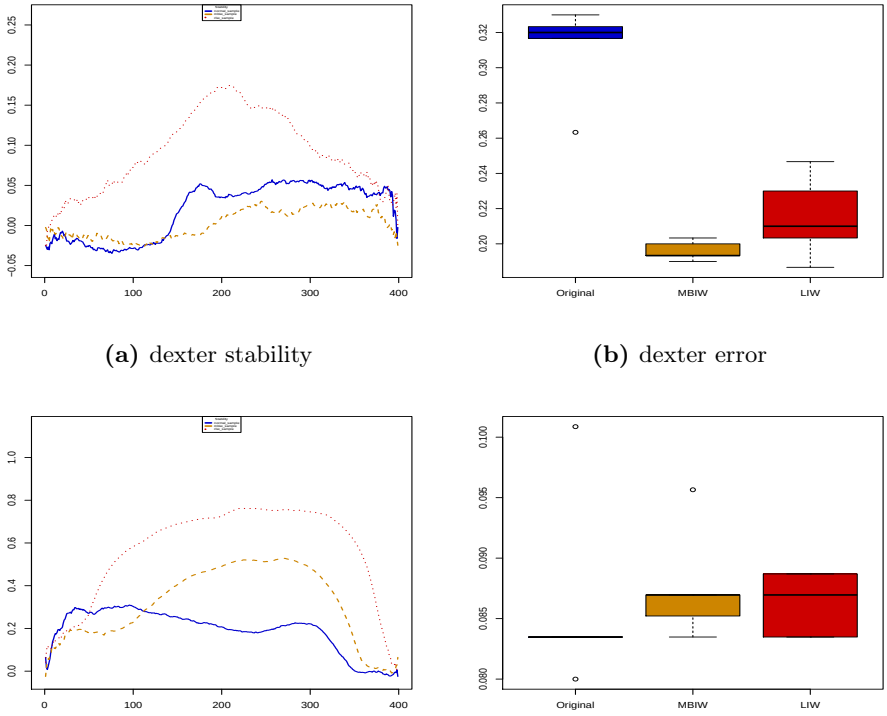



**(c)** madelon stability **(d)** madelon error




**(e)** gisette stability **(f)** gisette error

**Figure C.53:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



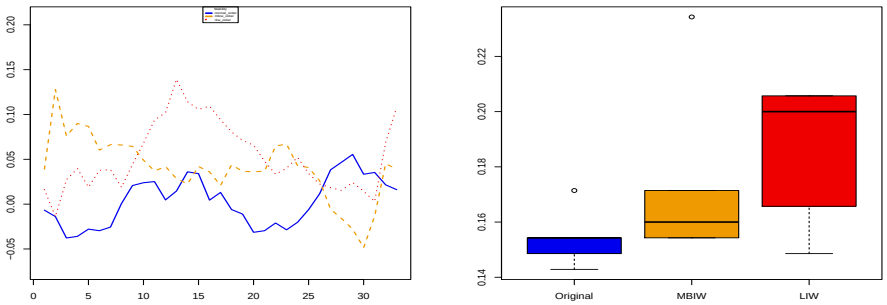

**(a)** dexter stability

**(b)** dexter error

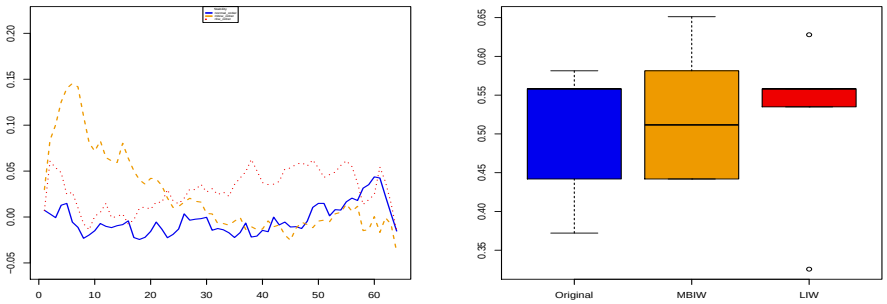
**(c)** dorothea stability

**(d)** dorothea error

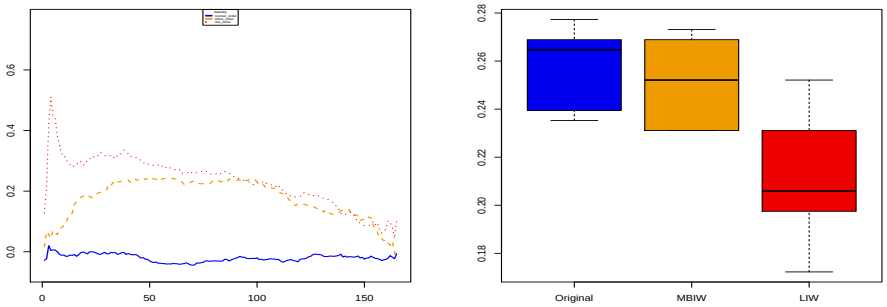
**Figure C.54:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.




**(a)** ionosphere stability **(b)** ionosphere error

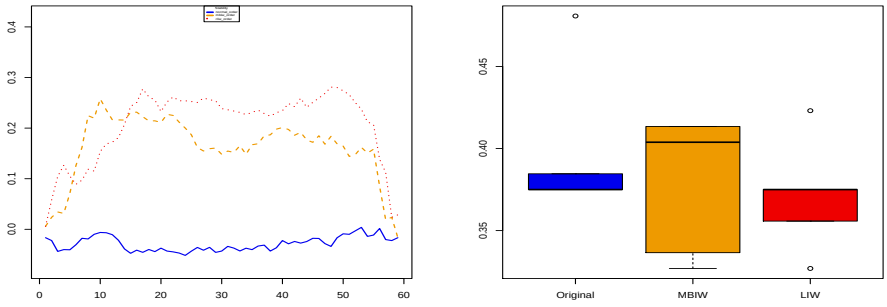



**(c)** mammogram stability **(d)** mammogram error




**(e)** musk stability **(f)** musk error

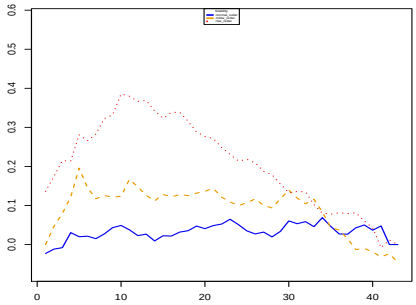
**Figure C.55:** Feature stability on UCI data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



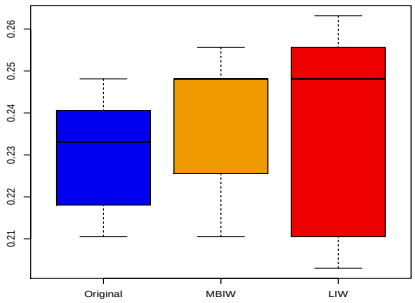

**(a)** sonar stability



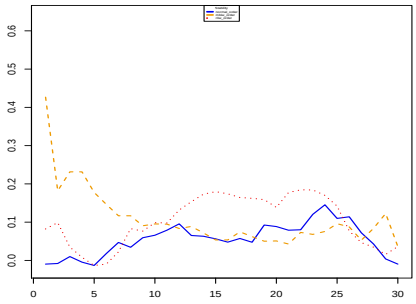

**(b)** sonar error



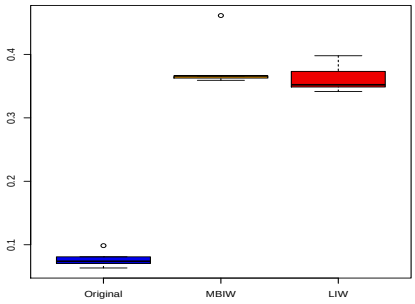

**(c)** spectf stability




**(d)** spectf error

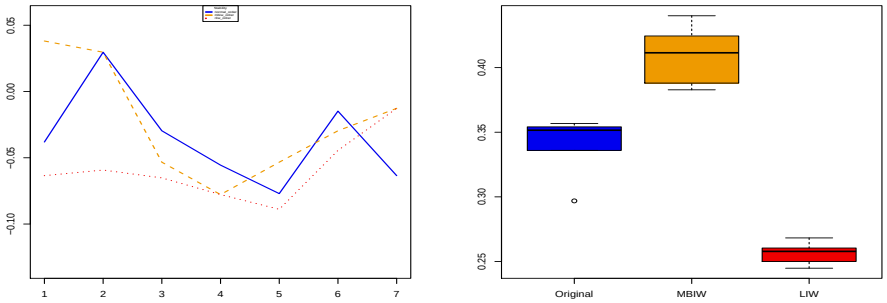



**(e)** wdbc stability



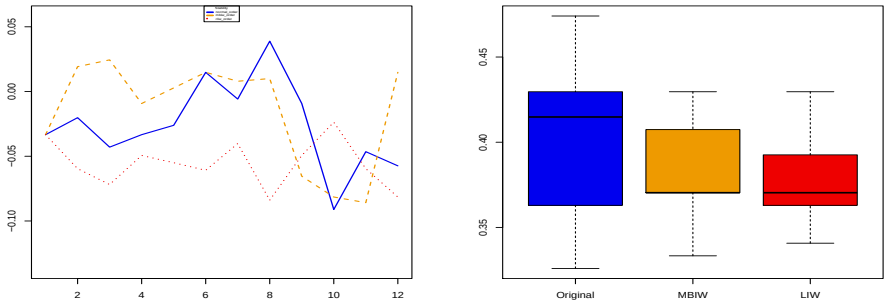

**(f)** wdbc error

**Figure C.56:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.




**(a)** diabetes stability **(b)** diabetes error

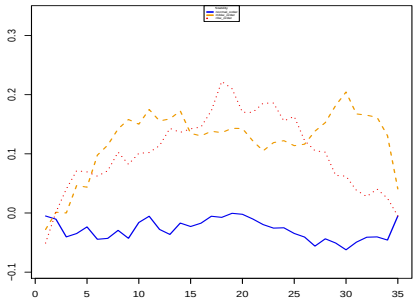
**Figure C.57:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



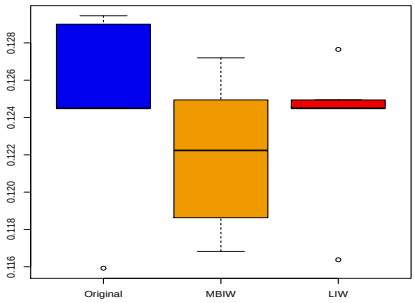

**(a)** heart statlog stability



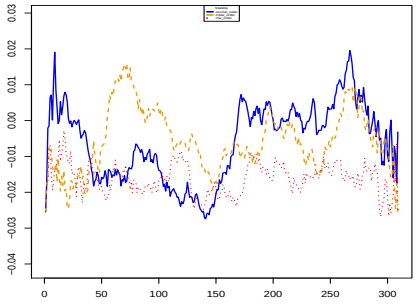

**(b)** heart statlog error



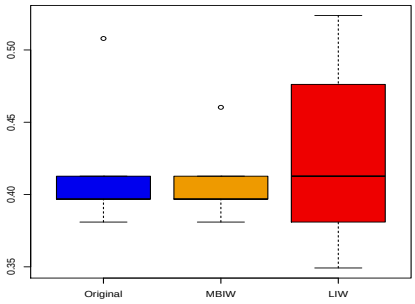

**(c)** landsat stability




**(d)** landsat error

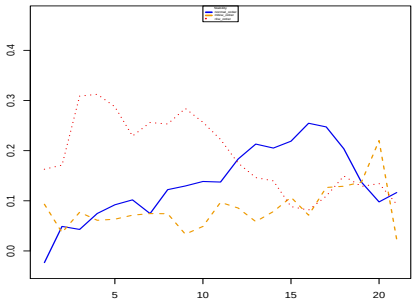



**(e)** lsvt voice stability

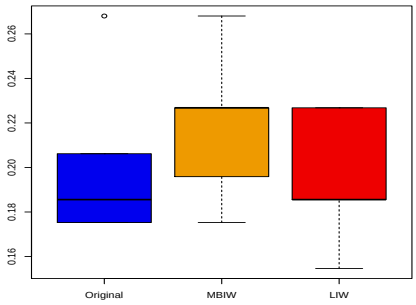



**(f)** lsvt voice error

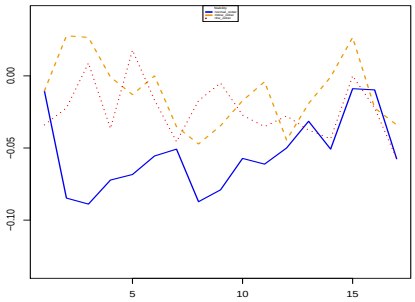
**Figure C.58:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



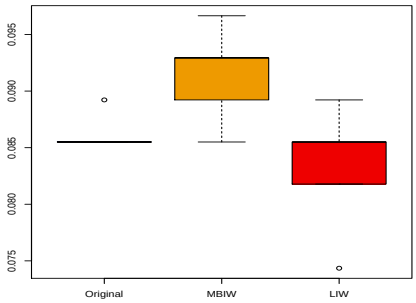

**(a)** parkinsons statlog stability



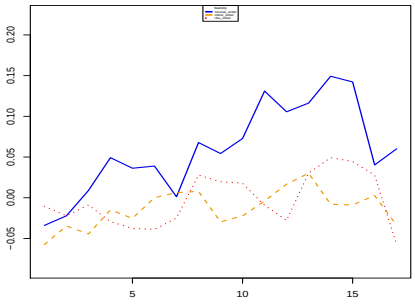

**(b)** parkinsons statlog error



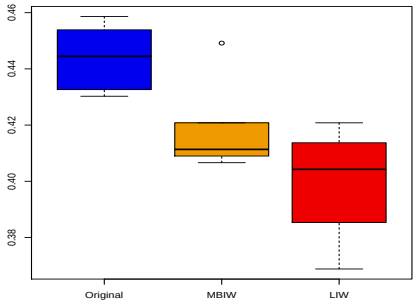

**(c)** pop failures stability




**(d)** pop failures error

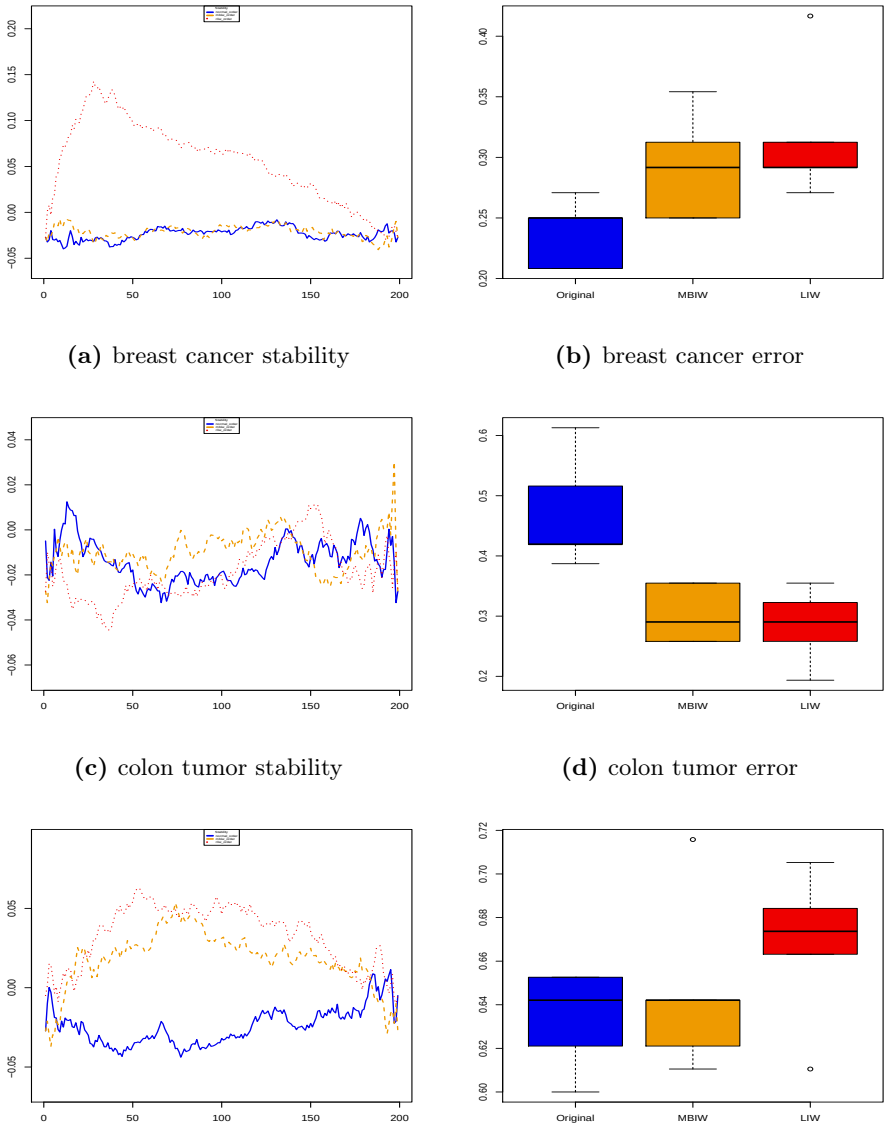



**(e)** vehicle stability




**(f)** vechicle voice error

**Figure C.59:** Feature stability on UCI data with $order\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.




**(a)** breast cancer stability

**(b)** breast cancer error

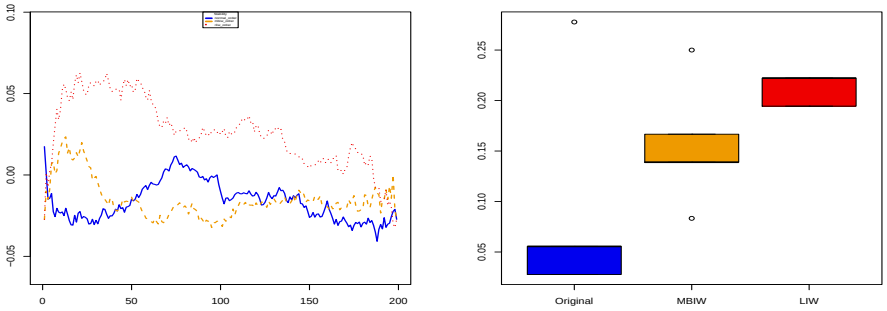
**(c)** colon tumor stability

**(d)** colon tumor error

**(e)** gcm stability

**(f)** gcm error

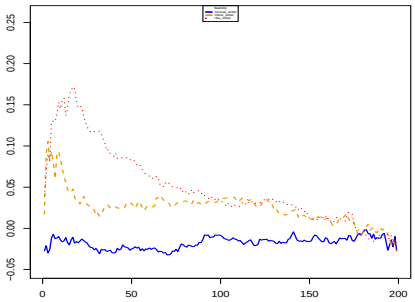
**Figure C.60:** Feature stability on microarray data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



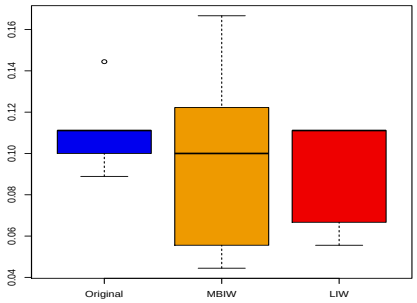

**(a)** leukemia stability



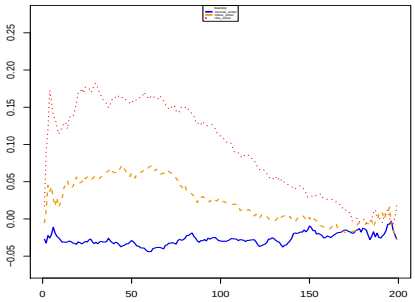

**(b)** leukemia error



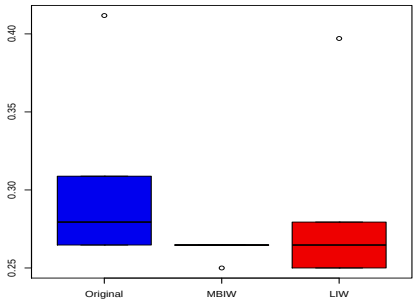

**(c)** lung cancer stability




**(d)** lung cancer error

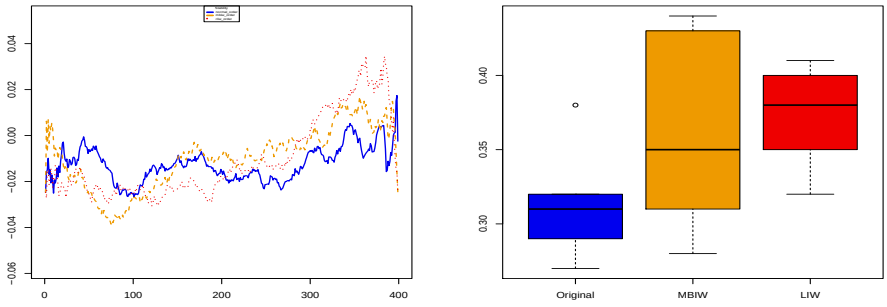



**(e)** prostate cancer stability




**(f)** prostate cancer error

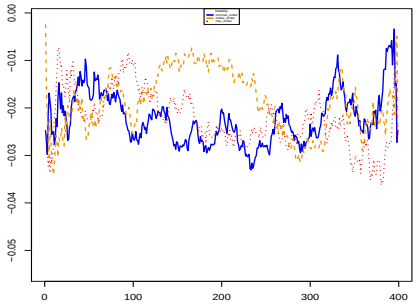
**Figure C.61:** Feature stability on microarray data with *order*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



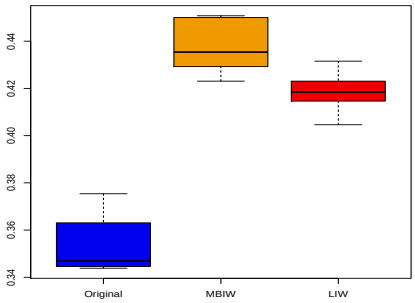

**(a)** arcene stability



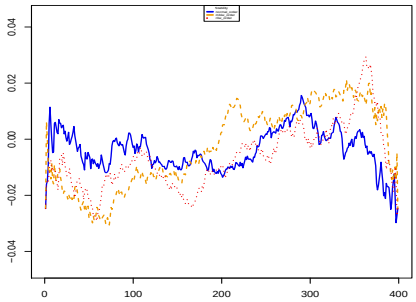

**(b)** arcene error



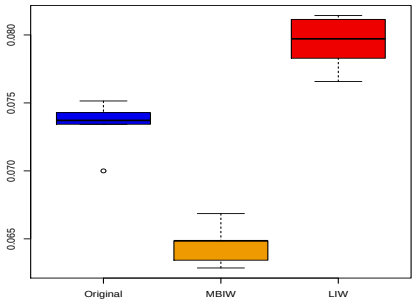

**(c)** madelon stability




**(d)** madelon error

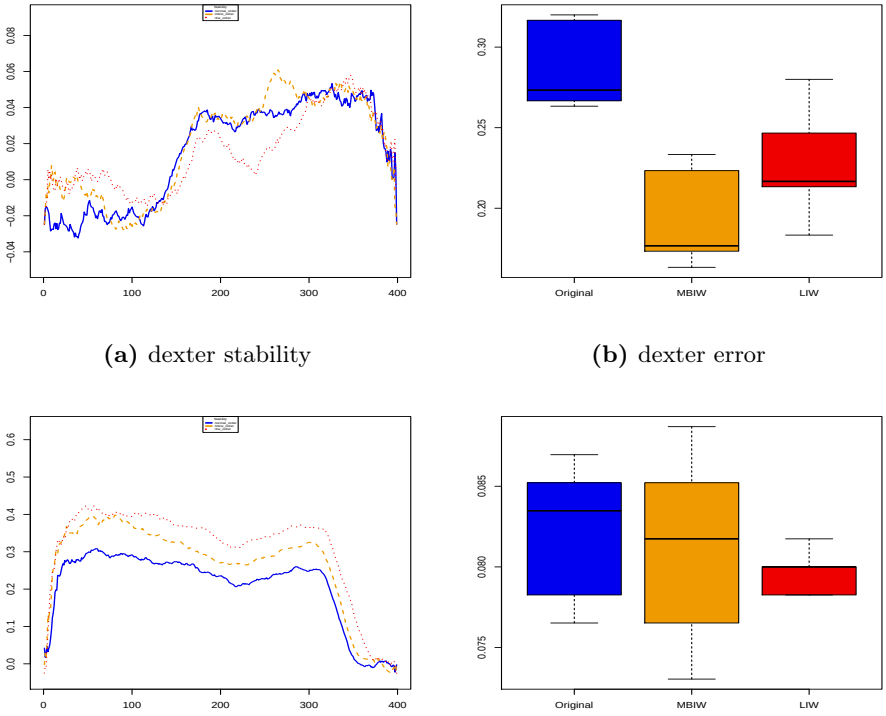



**(e)** gisette stability




**(f)** gisette error

**Figure C.62:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



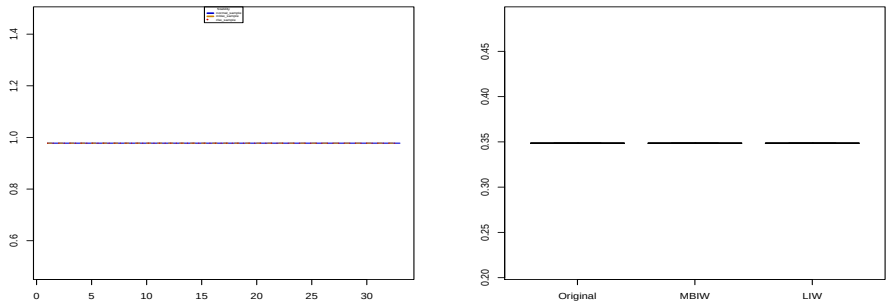

**(a)** dexter stability

**(b)** dexter error

**(c)** dorothea stability

**(d)** dorothea error

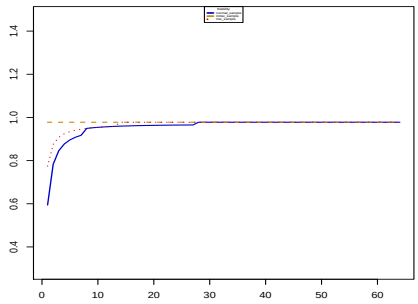
**Figure C.63:** Feature stability on NIPS Challenge data with $order\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **RandomForests**, the dashed line is the weighted **MBIW** + **RandomForests** version and the dotted one the weighted **RLIW** + **RandomForests** verion). Right plot shows the average test errors for **RandomForests**, **MBIW** + **RandomForests** and **RLIW** + **RandomForests** respectively.



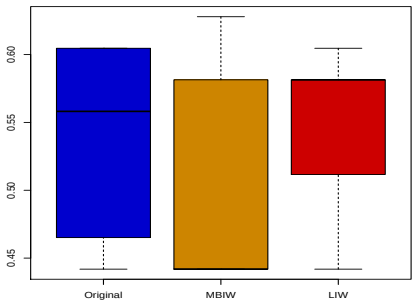

**(a)** ionosphere stability



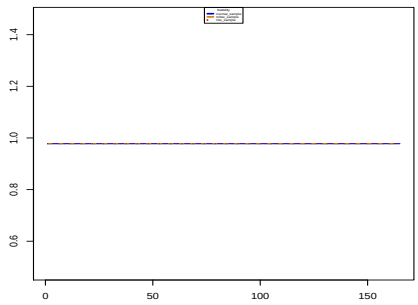

**(b)** ionosphere error



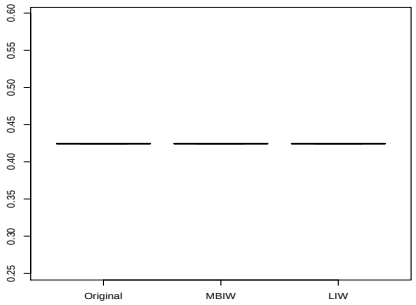

**(c)** mammogram stability




**(d)** mammogram error

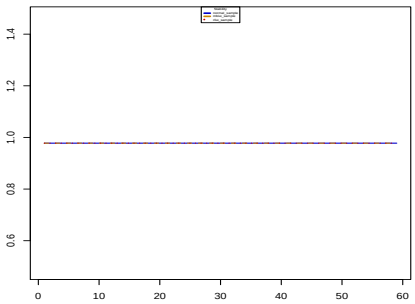



**(e)** musk stability

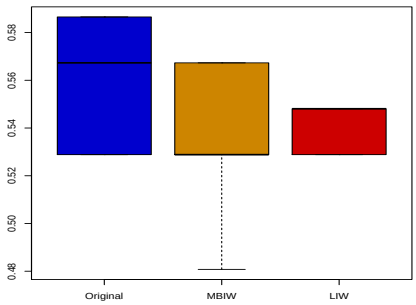



**(f)** musk error

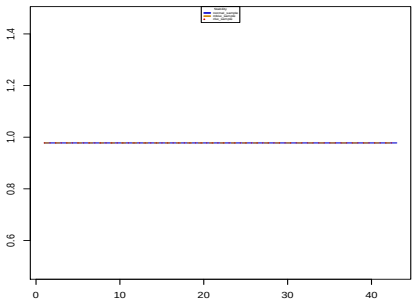
**Figure C.64:** Feature stability on UCI data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



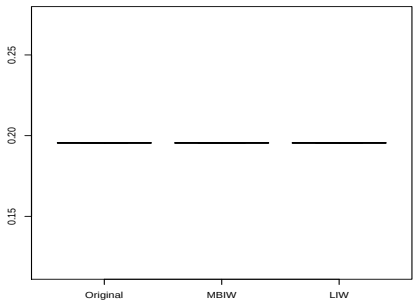

**(a)** sonar stability



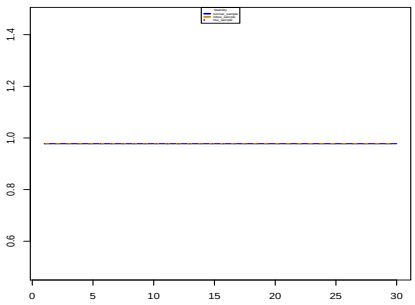

**(b)** sonar error



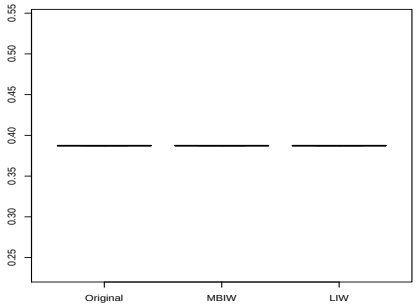

**(c)** spectf stability




**(d)** spectf error

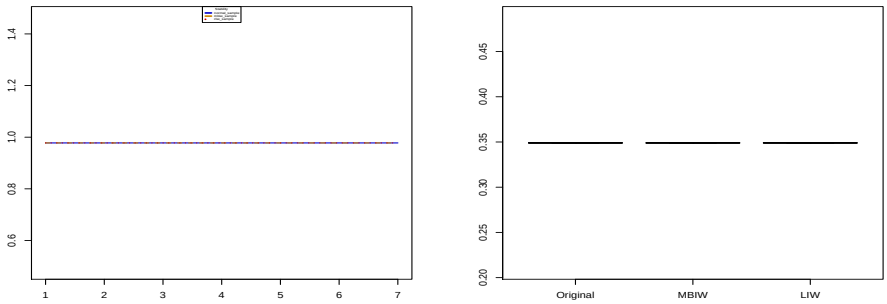



**(e)** wdbc stability




**(f)** wdbc error

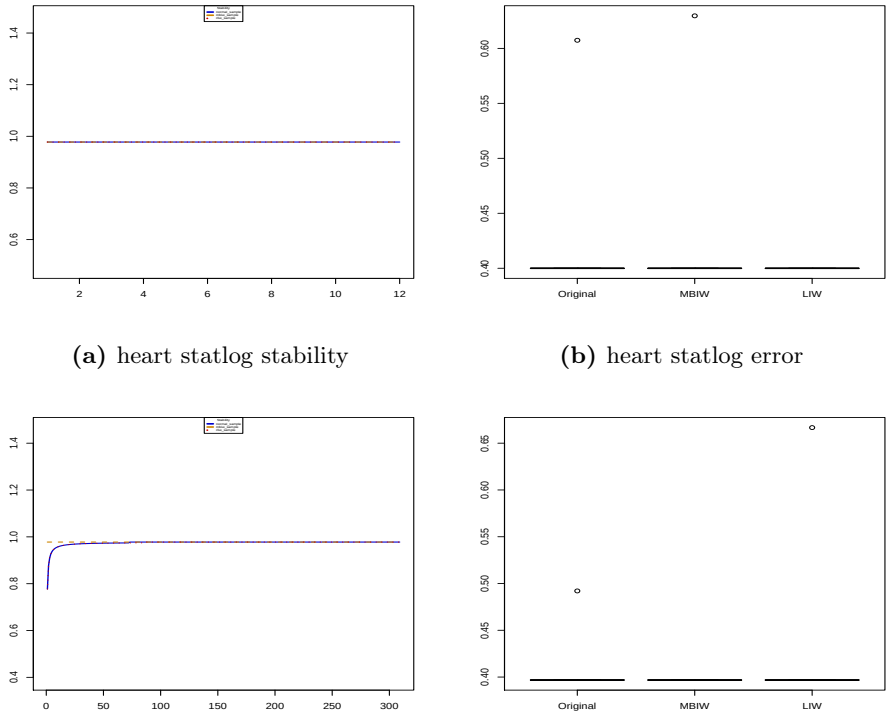
**Figure C.65:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.




**(a)** diabetes stability

**(b)** diabetes error

**Figure C.66:** Feature stability on UCI data with $sample\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



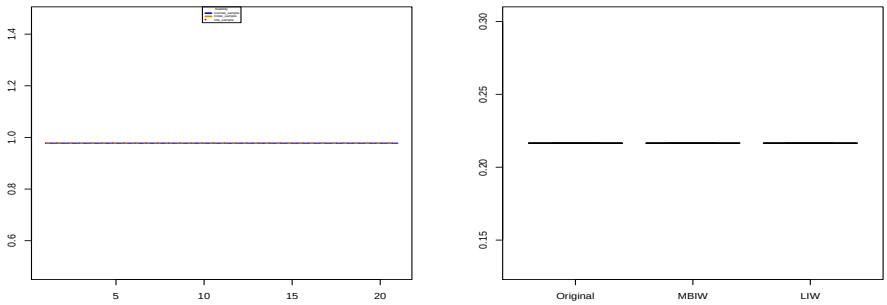

**(a)** heart statlog stability

**(b)** heart statlog error

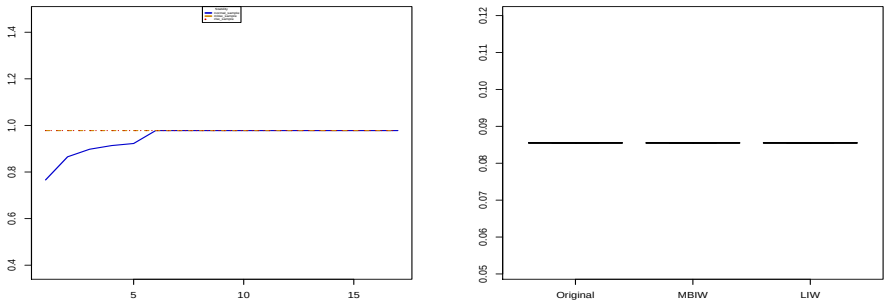
**(c)** lsvt voice stability

**(d)** lsvt voice error

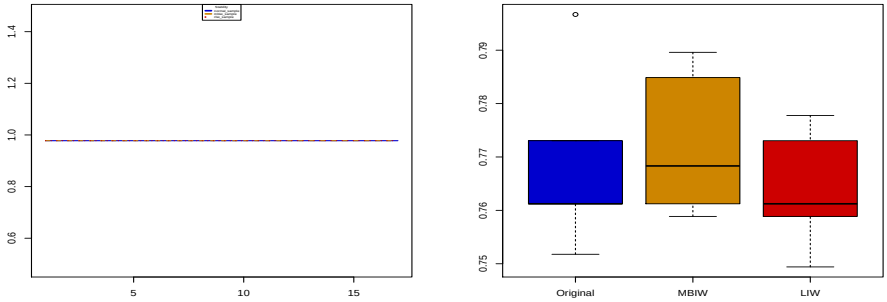
**Figure C.67:** Feature stability on UCI data with $sample\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.




**(a)** parkinsons statlog stability **(b)** parkinsons statlog error

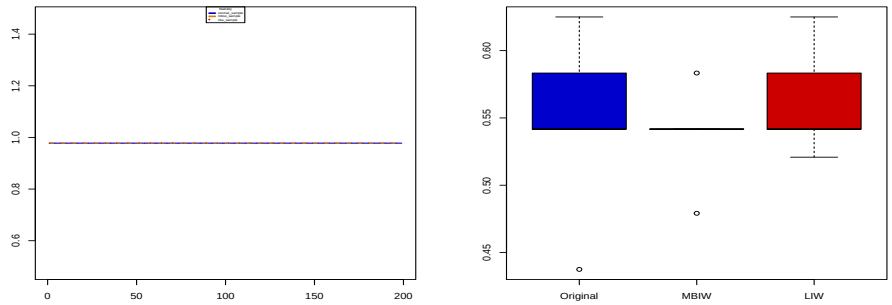



**(c)** pop failures stability **(d)** pop failures error

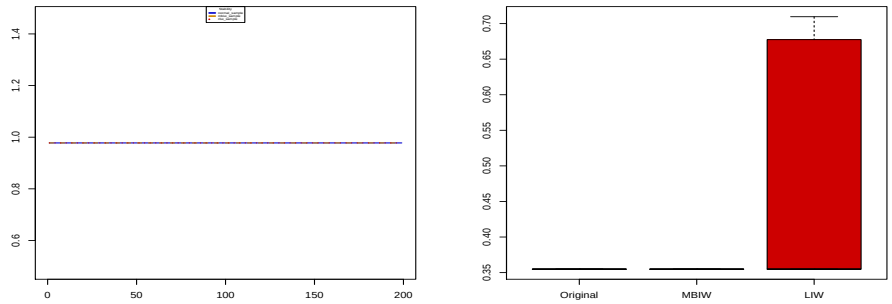



**(e)** vehicle stability **(f)** vechicle voice error

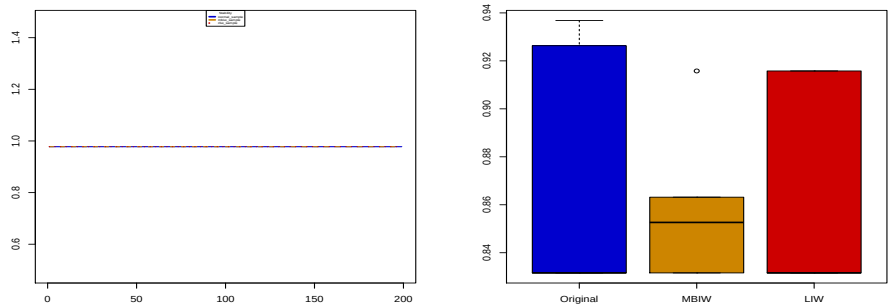
**Figure C.68:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.




**(a)** breast cancer stability **(b)** breast cancer error

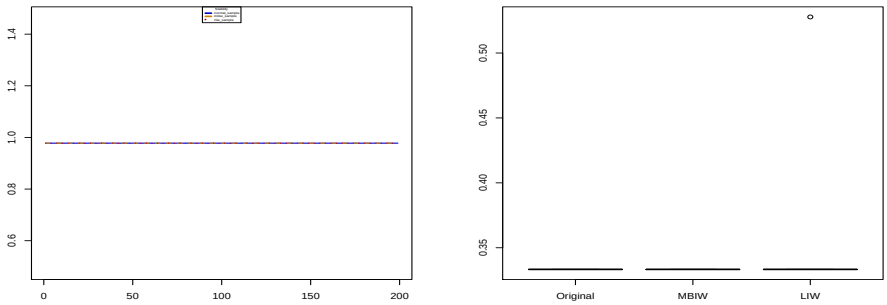



**(c)** colon tumor stability **(d)** colon tumor error




**(e)** gcm stability **(f)** gcm error

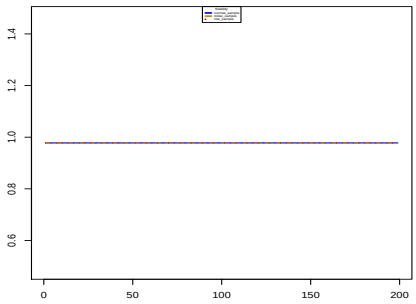
**Figure C.69:** Feature stability on microarray data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



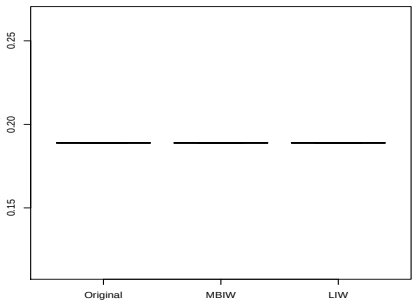

**(a)** leukemia stability



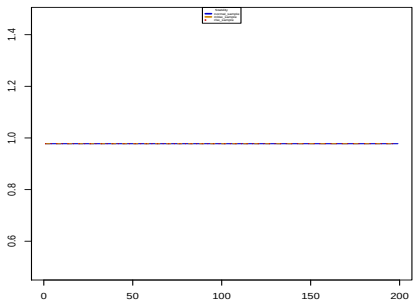

**(b)** leukemia error



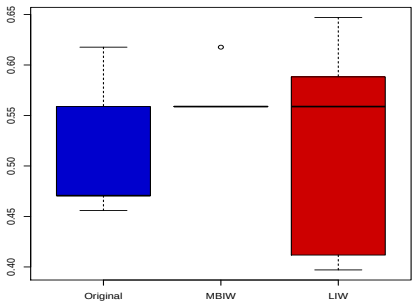

**(c)** lung cancer stability




**(d)** lung cancer error

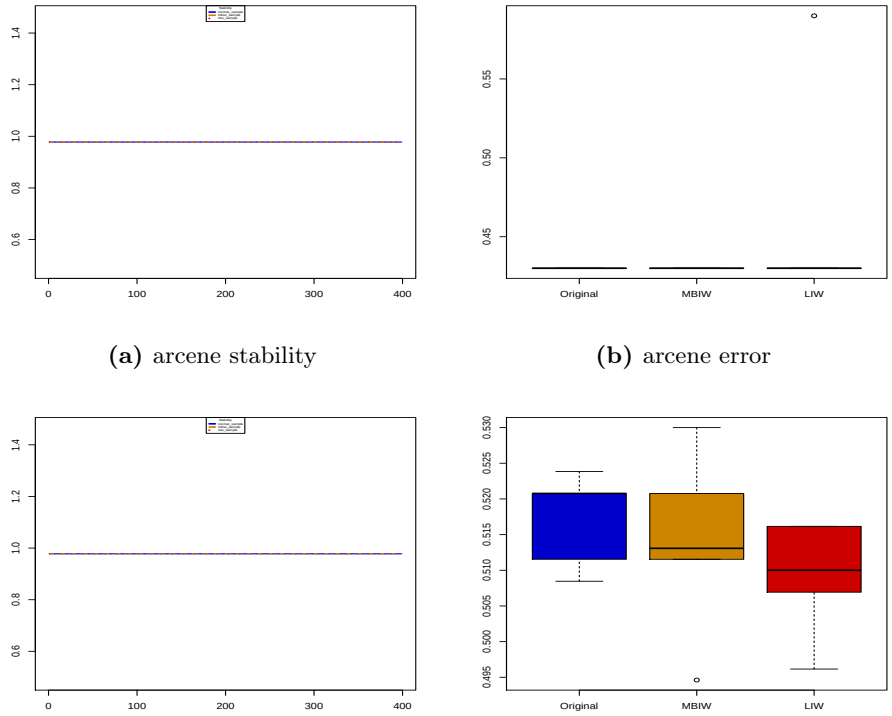



**(e)** prostate cancer stability




**(f)** prostate cancer error

**Figure C.70:** Feature stability on microarray data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



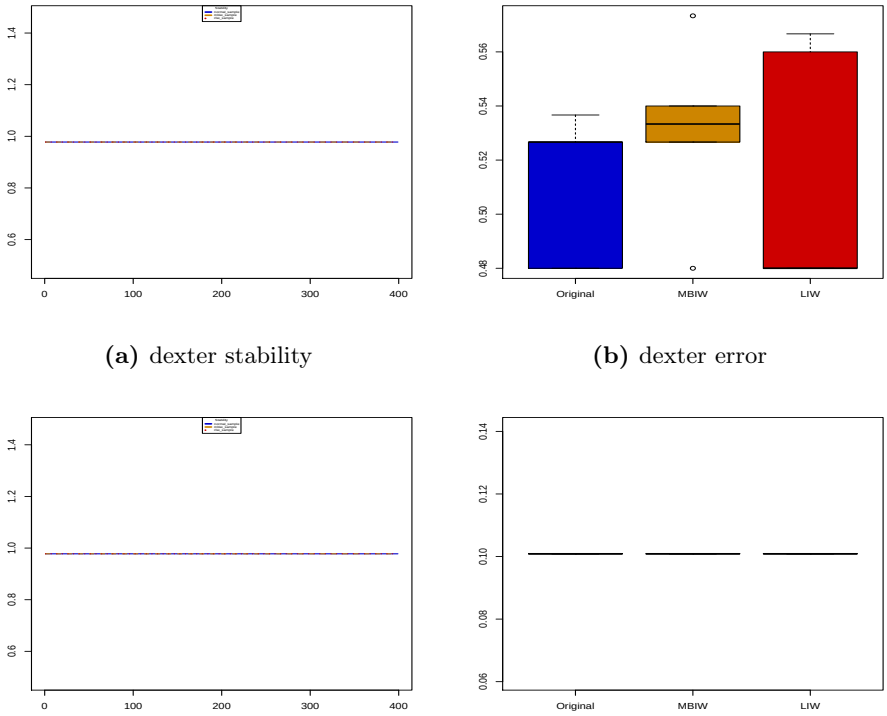

**(a)** arcene stability

**(b)** arcene error

**(c)** madelon stability

**(d)** madelon error

**Figure C.71:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



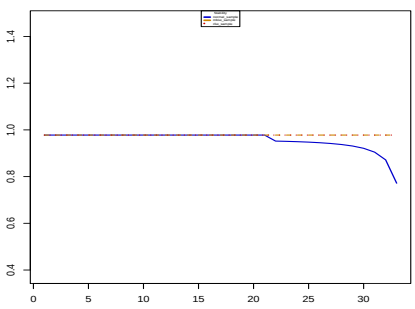

**(a)** dexter stability

**(b)** dexter error

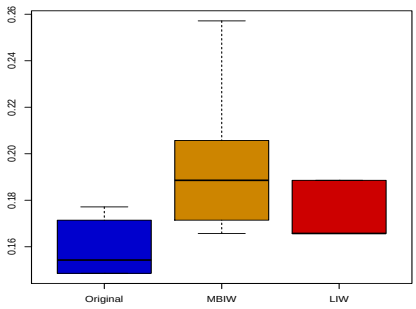
**(c)** dorothea stability

**(d)** dorothea error

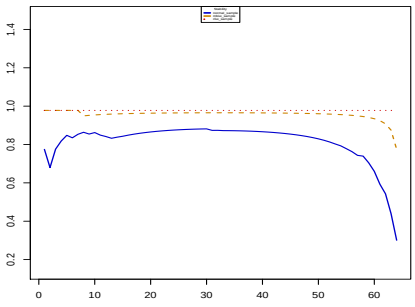
**Figure C.72:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **IG**, the dashed line is the weighted **MBIW** + **InformationGain** version and the dotted one the weighted **RLIW** + **InformationGain** verion). Right plot shows the average test errors for **IG**, **MBIW** + **InformationGain** and **RLIW** + **InformationGain** respectively.



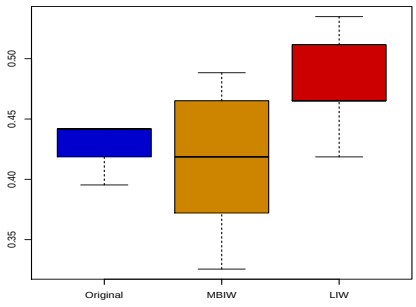
**(a)** ionosphere stability



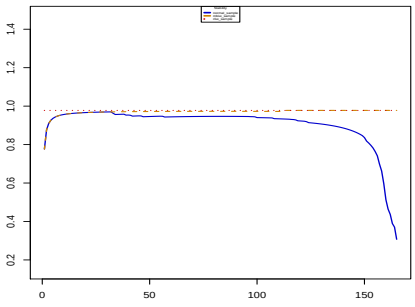

**(b)** ionosphere error



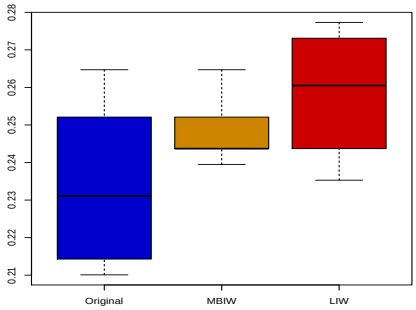
**(c)** mammogram stability




**(d)** mammogram error

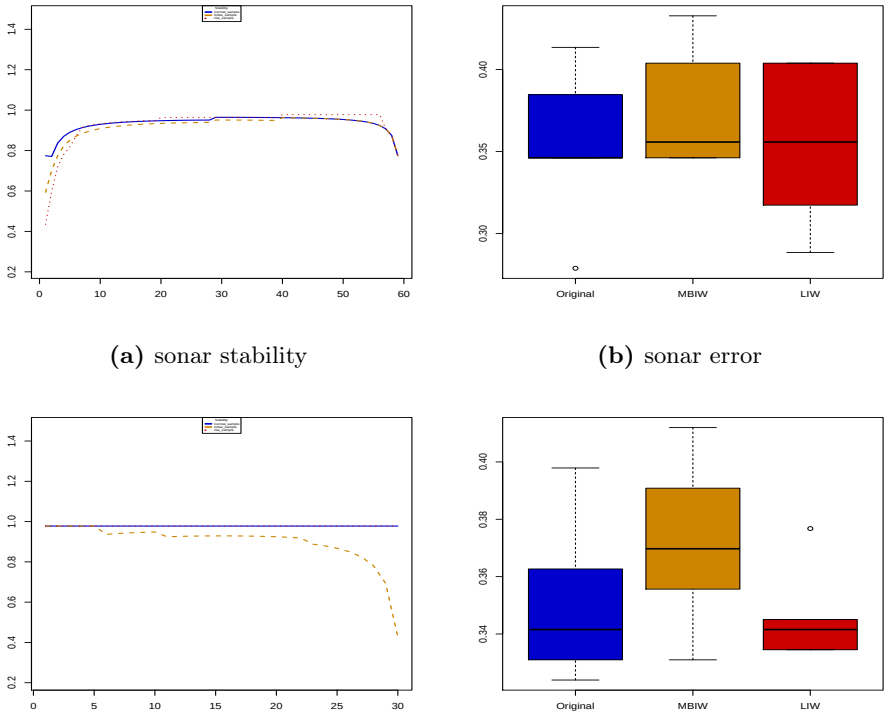


**(e)** musk stability




**(f)** musk error

**Figure C.73:** Feature stability on UCI data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.



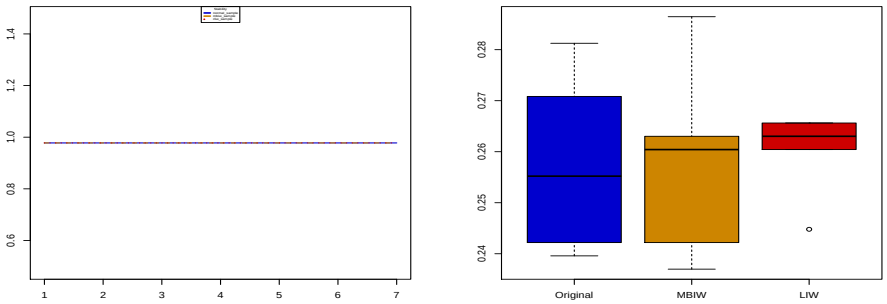

**(a)** sonar stability

**(b)** sonar error

**(c)** wdbc stability

**(d)** wdbc error

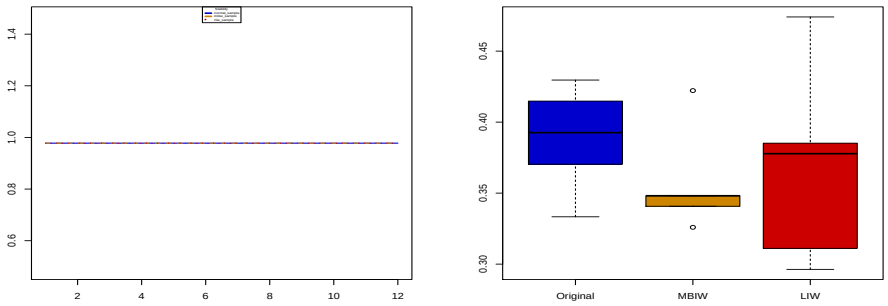
**Figure C.74:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.




**(a)** diabetes stability

**(b)** diabetes error

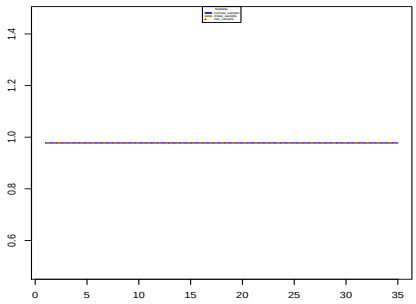
**Figure C.75:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.



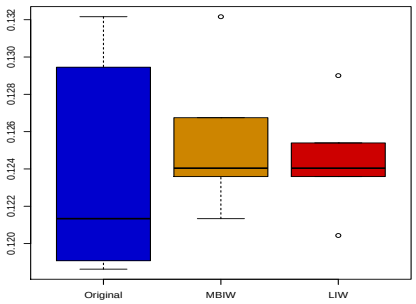

**(a)** heart statlog stability



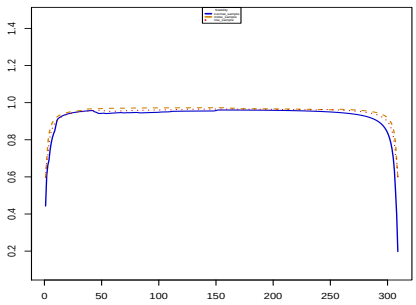

**(b)** heart statlog error



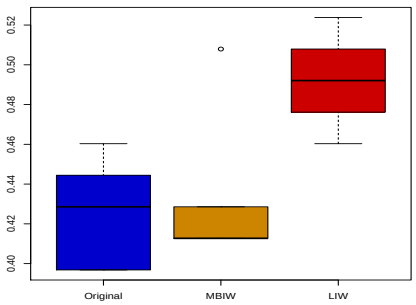

**(c)** landsat stability




**(d)** landsat error

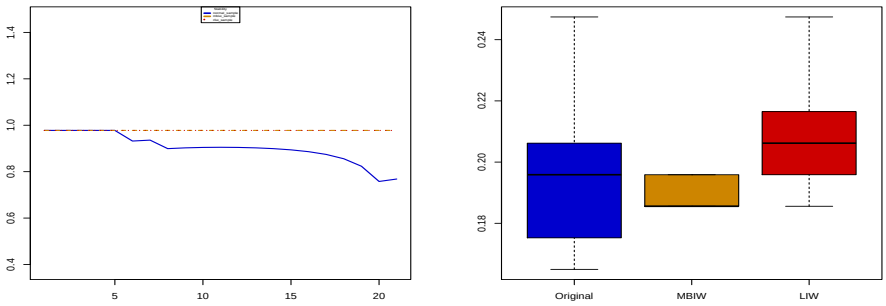



**(e)** lsvt voice stability

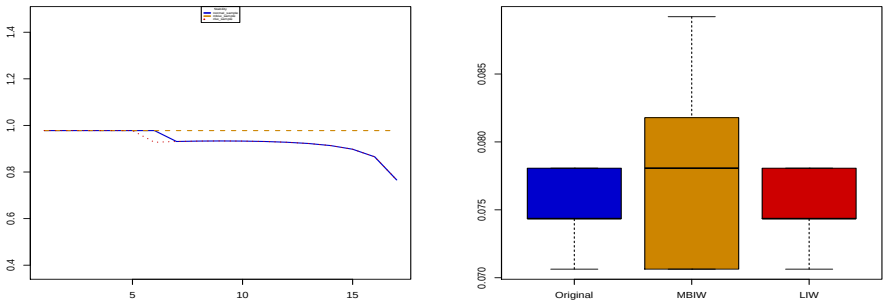



**(f)** lsvt voice error

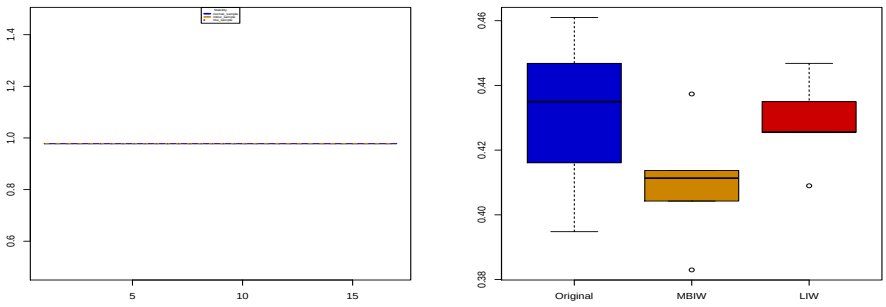
**Figure C.76:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.




**(a)** parkinsons statlog stability **(b)** parkinsons statlog error

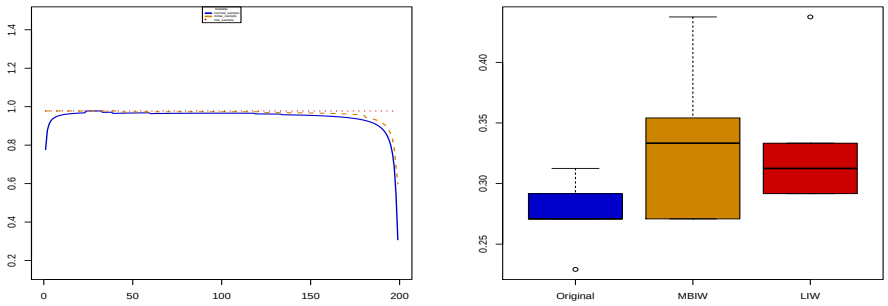



**(c)** pop failures stability **(d)** pop failures error




**(e)** vehicle stability **(f)** vechicle voice error

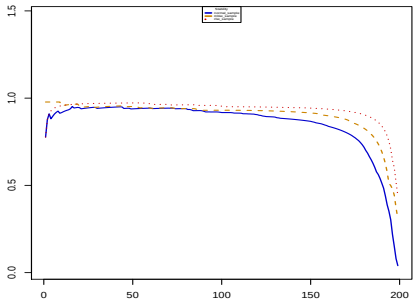
**Figure C.77:** Feature stability on UCI data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.



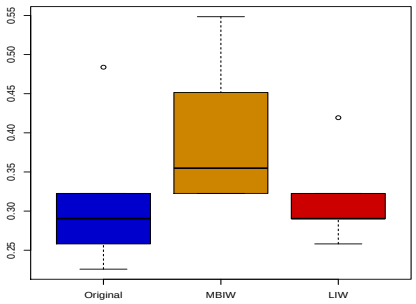

**(a)** breast cancer stability



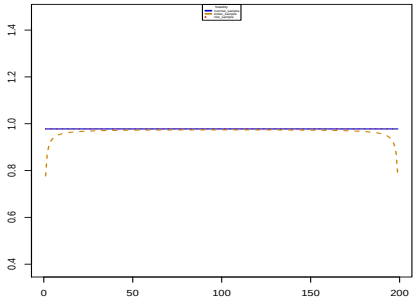

**(b)** breast cancer error



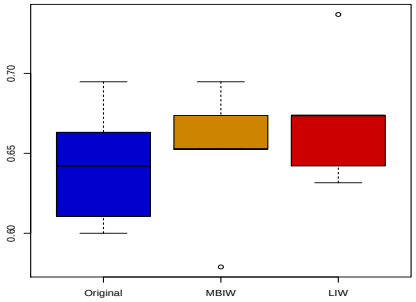

**(c)** colon tumor stability




**(d)** colon tumor error

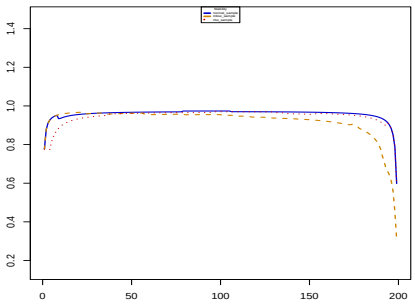



**(e)** gcm stability

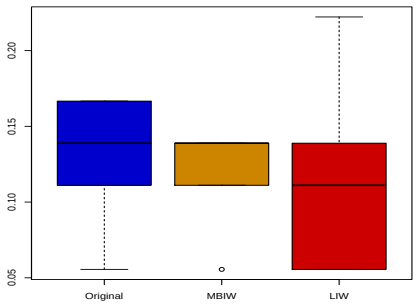



**(f)** gcm error

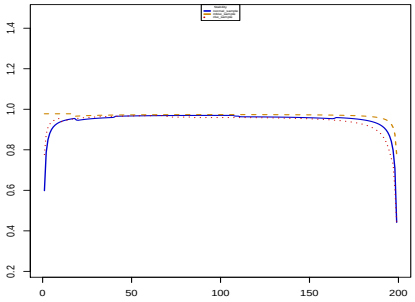
**Figure C.78:** Feature stability on microarray data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.



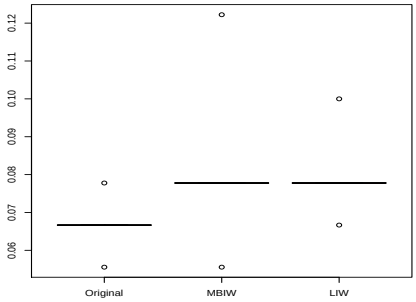

**(a)** leukemia stability



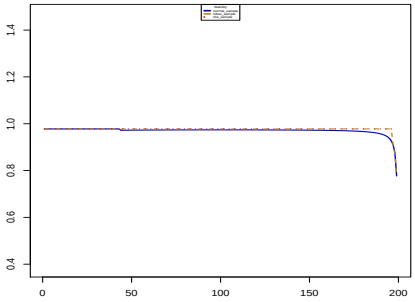

**(b)** leukemia error



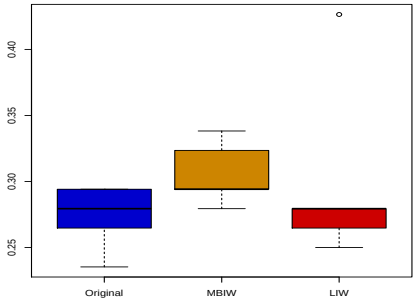

**(c)** lung cancer stability




**(d)** lung cancer error

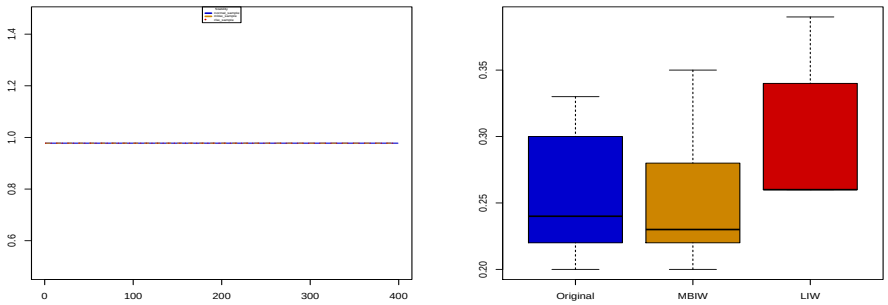



**(e)** prostate cancer stability




**(f)** prostate cancer error

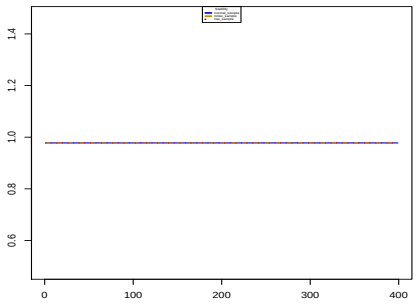
**Figure C.79:** Feature stability on microarray data with *sample*$\Delta$ combination (continued). Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.



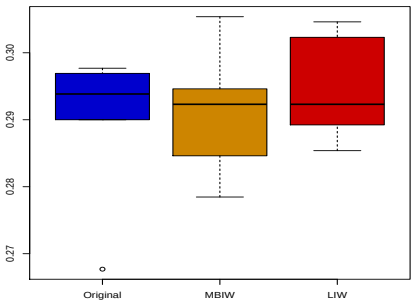

**(a)** arcene stability



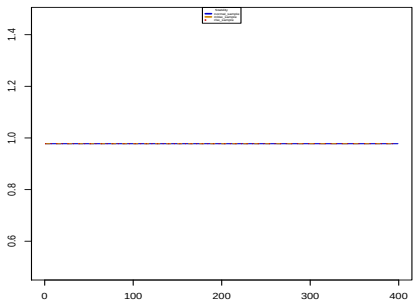

**(b)** arcene error



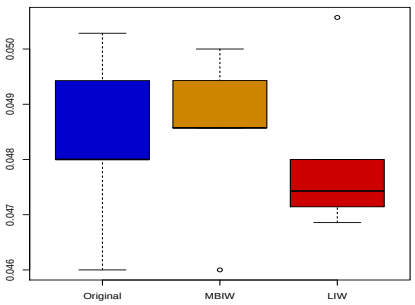

**(c)** madelon stability




**(d)** madelon error

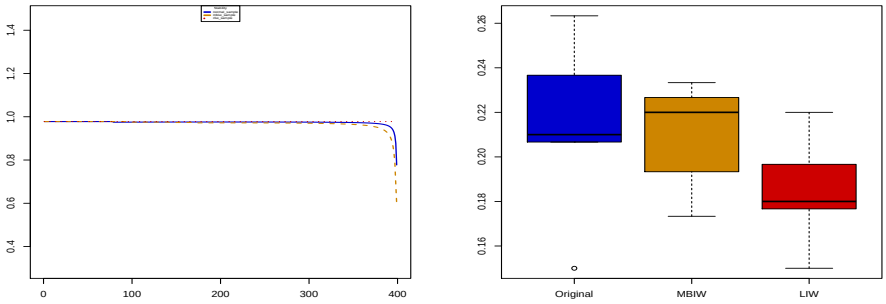



**(e)** gisette stability



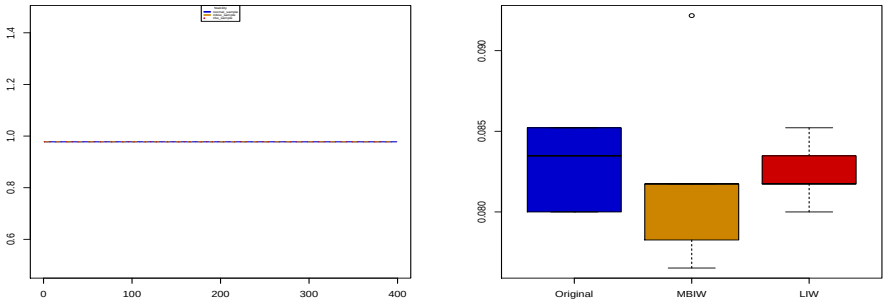

**(f)** gisette error

**Figure C.80:** Feature stability on NIPS Challenge data with *sample*$\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.




**(a)** dexter stability

**(b)** dexter error




**(c)** dorothea stability

**(d)** dorothea error

**Figure C.81:** Feature stability on NIPS Challenge data with $sample\Delta$ combination. Left plots show average $S_{Kuncheva}$ against subset size (the bold line is the normal **1R**, the dashed line is the weighted **MBIW** + **1R** version and the dotted one the weighted **RLIW** + **1R** verion). Right plot shows the average test errors for **1R**, **MBIW** + **1R** and **RLIW** + **1R** respectively.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.4735±0.1505 | 0.4211±0.1563 | 0.4688 |
| glass | 0.2136±0.1335 | 0.2105±0.0935 | 0.9326 |
| heart_statlog | 0.2945±0.1499 | **0.4078±0.2121** | 0.021 |
| ionosphere | **0.261±0.1223** | 0.1465±0.0451 | 0 |
| landsat_train | **0.6044±0.1132** | 0.2809±0.1325 | 0 |
| lsvt_voice | **0.2963±0.1683** | 0.2289±0.262 | 0 |
| mammogram | **0.2169±0.0973** | 0.1747±0.1043 | 0.0155 |
| musk | **0.2465±0.067** | 0.1043±0.0479 | 0 |
| parkinsons | **0.4871±0.165** | 0.3287±0.2138 | 0 |
| pop_failures | **0.3754±0.2649** | 0.2488±0.2397 | 0.0013 |
| sonar | **0.3962±0.1487** | 0.3767±0.1624 | 0.0088 |
| spectf | **0.2545±0.11** | 0.0804±0.0551 | 0 |
| vehicle | 0.489±0.1673 | 0.5324±0.172 | 0.7119 |
| waveform | **0.5401±0.0927** | 0.2572±0.0571 | 0 |
| wdbc | **0.7246±0.1852** | 0.3758±0.3204 | 0 |

**Table C.1:** Summary of stability results on UCI datasets for **SimbaMiw** with *normal*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2688±0.0242 | 0.2922±0.0178 | 0.2205 |
| glass | 0.4037±0.0514 | 0.4262±0.0377 | 0.5501 |
| heart_statlog | 0.3437±0.0365 | 0.357±0.0122 | 0.3134 |
| ionosphere | 0.1863±0.0312 | 0.2354±0.0281 | 0.067 |
| landsat_train | **0.1243±0.0014** | 0.1425±0.0049 | 5e-04 |
| lsvt_voice | 0.4349±0.0329 | 0.4317±0.0468 | 0.8541 |
| mammogram | 0.507±0.0921 | 0.4558±0.0353 | 0.3455 |
| musk | **0.2681±0.0314** | 0.3429±0.0258 | 0.0134 |
| parkinsons | 0.1959±0.0193 | 0.2021±0.0548 | 0.8223 |
| pop_failures | 0.0796±0.0068 | **0.0639±0.0048** | 0.0146 |
| sonar | 0.3615±0.0357 | 0.3212±0.0422 | 0.2306 |
| spectf | 0.2586±0.0544 | 0.1955±0.0323 | 0.0626 |
| vehicle | **0.4203±0.0181** | 0.5348±0.0162 | 1e-04 |
| waveform | **0.1742±0.0035** | 0.2106±0.0043 | 1e-04 |
| wdbc | 0.3789±0.0101 | 0.4014±0.019 | 0.0899 |

**Table C.2:** Summary of classification error results on UCI datasets for **SimbaMiw** with *normal*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1786±0.0387** | 0.1268±0.0296 | 0 |
| ma_colon_tumor | **0.1919±0.0859** | 0.1364±0.0811 | 0 |
| ma_gcm | **0.4216±0.1069** | 0.2859±0.1738 | 0 |
| ma_leukemia | **0.3004±0.1188** | 0.1654±0.0866 | 0 |
| ma_lung_cancer | **0.52±0.0594** | 0.372±0.0973 | 0 |
| ma_prostate_cancer | **0.3805±0.1496** | 0.1521±0.1549 | 0 |

**Table C.3:** Summary of stability results on Microarray datasets for **SimbaMiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.2875±0.0632 | 0.2917±0.0329 | 0.9181 |
| ma_colon_tumor | **0.3032±0.0489** | 0.4258±0.053 | 0.0173 |
| ma_gcm | 0.6547±0.0272 | 0.6632±0.0341 | 0.6885 |
| ma_leukemia | **0.0833±0.0393** | 0.2167±0.0304 | 0.0109 |
| ma_lung_cancer | 0.0578±0.0372 | 0.06±0.0186 | 0.9142 |
| ma_prostate_cancer | 0.2529±0.0554 | 0.2794±0.0453 | 0.2205 |

**Table C.4:** Summary of classification error results on Microarray datasets for **SimbaMiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.3073±0.1014** | 0.144±0.0912 | 0 |
| dexter | **0.2903±0.0849** | 0.2299±0.0674 | 0 |
| dorothea | **0.1685±0.0597** | 0.1218±0.0384 | 0 |
| gisette | **0.5947±0.0577** | 0.5061±0.13 | 0 |
| madelon | 0.3067±0.1575 | **0.6993±0.182** | 0 |

**Table C.5:** Summary of stability results on NIPS datasets for **SimbaMiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.268±0.0665 | 0.376±0.0518 | 0.0928 |
| dexter | 0.2387±0.0565 | 0.1887±0.0183 | 0.0995 |
| dorothea | 0.0814±0.0031 | 0.0835±0.0059 | 0.4581 |
| gisette | **0.0478±0.002** | 0.0552±0.0031 | 0.022 |
| madelon | 0.2929±0.0096 | **0.1371±0.0032** | 0 |

**Table C.6:** Summary of classification error results on NIPS datasets for **SimbaMiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.464±0.151 | 0.4211±0.1563 | 0.4017 |
| glass | **0.3429±0.1951** | 0.2105±0.0935 | 0.0249 |
| heart_statlog | 0.295±0.2774 | 0.4078±0.2121 | 0.0923 |
| ionosphere | **0.1914±0.0599** | 0.1465±0.0451 | 0 |
| landsat_train | 0.2428±0.1171 | **0.2809±0.1325** | 0.0011 |
| lsvt_voice | **0.2695±0.2546** | 0.2289±0.262 | 0 |
| mammogram | 0.0644±0.1063 | **0.1747±0.1043** | 0 |
| musk | 0.1084±0.0337 | 0.1043±0.0479 | 0.0688 |
| parkinsons | 0.3507±0.1651 | 0.3287±0.2138 | 0.2157 |
| pop_failures | 0.1762±0.1737 | **0.2488±0.2397** | 2e-04 |
| sonar | 0.3283±0.1586 | **0.3767±0.1624** | 0 |
| spectf | 0.0927±0.0896 | 0.0804±0.0551 | 0.6625 |
| vehicle | 0.4233±0.1087 | **0.5324±0.172** | 0.0041 |
| waveform | 0.2218±0.0872 | 0.2572±0.0571 | 0.0613 |
| wdbc | 0.2205±0.258 | **0.3758±0.3204** | 1e-04 |

**Table C.7:** Summary of stability results on UCI datasets for **SimbaLiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2719±0.0201 | 0.2922±0.0178 | 0.1847 |
| glass | 0.4598±0.0589 | 0.4262±0.0377 | 0.4537 |
| heart_statlog | 0.3689±0.0632 | 0.357±0.0122 | 0.6936 |
| ionosphere | 0.2411±0.0237 | 0.2354±0.0281 | 0.6164 |
| landsat_train | 0.1928±0.0036 | **0.1425±0.0049** | 0 |
| lsvt_voice | 0.4476±0.0411 | 0.4317±0.0468 | 0.3262 |
| mammogram | **0.2884±0.0453** | 0.4558±0.0353 | 0.0011 |
| musk | 0.316±0.0261 | 0.3429±0.0258 | 0.0682 |
| parkinsons | 0.1938±0.0402 | 0.2021±0.0548 | 0.7611 |
| pop_failures | 0.0699±0.0085 | 0.0639±0.0048 | 0.2272 |
| sonar | **0.2462±0.0364** | 0.3212±0.0422 | 0.0295 |
| spectf | 0.2316±0.0187 | 0.1955±0.0323 | 0.0584 |
| vehicle | **0.4525±0.0171** | 0.5348±0.0162 | 0.0013 |
| waveform | 0.2178±0.0057 | 0.2106±0.0043 | 0.1408 |
| wdbc | 0.3817±0.0192 | 0.4014±0.019 | 0.0604 |

**Table C.8:** Summary of classification error results on UCI datasets for **SimbaLiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1763±0.0212** | 0.1268±0.0296 | 0 |
| ma_colon_tumor | **0.2313±0.144** | 0.1364±0.0811 | 0 |
| ma_gcm | **0.3126±0.2065** | 0.2859±0.1738 | 0 |
| ma_leukemia | **0.2067±0.0971** | 0.1654±0.0866 | 0 |
| ma_lung_cancer | **0.3846±0.1029** | 0.372±0.0973 | 2e-04 |
| ma_prostate_cancer | **0.2028±0.161** | 0.1521±0.1549 | 0 |

**Table C.9:** Summary of stability results on Microarray datasets for **SimbaLiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3417±0.0891 | 0.2917±0.0329 | 0.3624 |
| ma_colon_tumor | 0.3677±0.0629 | 0.4258±0.053 | 0.2954 |
| ma_gcm | 0.6695±0.0354 | 0.6632±0.0341 | 0.8194 |
| ma_leukemia | **0.1167±0.0232** | 0.2167±0.0304 | 0.0061 |
| ma_lung_cancer | 0.06±0.0099 | 0.06±0.0186 | 1 |
| ma_prostate_cancer | 0.2971±0.0366 | 0.2794±0.0453 | 0.5473 |

**Table C.10:** Summary of classification error results on Microarray datasets for **SimbaLiw** with $normal\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.1498±0.1109** | 0.144±0.0912 | 0 |
| dexter | **0.2978±0.0597** | 0.2299±0.0674 | 0 |
| dorothea | **0.2159±0.0795** | 0.1218±0.0384 | 0 |
| gisette | 0.4746±0.1558 | **0.5061±0.13** | 0 |
| madelon | 0.5986±0.1595 | **0.6993±0.182** | 0 |

**Table C.11:** Summary of stability results on NIPS datasets for **SimbaLiw** with *normal*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.322±0.013 | 0.376±0.0518 | 0.0781 |
| dexter | 0.1913±0.0146 | 0.1887±0.0183 | 0.8361 |
| dorothea | 0.0856±0.0023 | 0.0835±0.0059 | 0.5012 |
| gisette | 0.0655±0.0032 | **0.0552±0.0031** | 1e-04 |
| madelon | 0.1395±0.006 | 0.1371±0.0032 | 0.1993 |

**Table C.12:** Summary of classification error results on NIPS datasets for **SimbaLiw** with *normal*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2776±0.159 | 0.3663±0.1533 | 0.0781 |
| glass | 0.1579±0.1245 | 0.2557±0.1408 | 0.1083 |
| heart_statlog | 0.3066±0.1443 | 0.277±0.2032 | 0.3394 |
| ionosphere | **0.2433±0.1152** | 0.0916±0.0336 | 0 |
| landsat_train | **0.5319±0.0999** | 0.2851±0.1477 | 0 |
| lsvt_voice | 0.2435±0.1838 | 0.2318±0.2645 | 0.1949 |
| mammogram | **0.1531±0.0913** | 0.0685±0.0591 | 0 |
| musk | **0.1746±0.0415** | 0.1047±0.0557 | 0 |
| parkinsons | **0.4434±0.1757** | 0.243±0.2157 | 0 |
| pop_failures | 0.3175±0.2283 | 0.2806±0.2492 | 0.1454 |
| sonar | **0.3308±0.164** | 0.2899±0.1619 | 0 |
| spectf | **0.1452±0.0695** | 0.1269±0.0994 | 0.0063 |
| vehicle | 0.4222±0.2121 | 0.4704±0.216 | 0.3778 |
| waveform | **0.4254±0.0563** | 0.2133±0.0702 | 0 |
| wdbc | **0.6514±0.204** | 0.3769±0.32 | 0 |

**Table C.13:** Summary of stability results on UCI datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | **0.2604±0.0137** | 0.2911±0.0162 | 0.0338 |
| glass | 0.4112±0.0391 | 0.4879±0.0829 | 0.0526 |
| heart_statlog | 0.3733±0.0634 | 0.3378±0.0358 | 0.2098 |
| ionosphere | **0.1691±0.0154** | 0.2549±0.0138 | 5e-04 |
| landsat_train | **0.1222±0.0047** | 0.145±0.0063 | 4e-04 |
| lsvt_voice | 0.4349±0.0329 | 0.4317±0.0468 | 0.8541 |
| mammogram | 0.5395±0.0861 | **0.4186±0.0717** | 0.0082 |
| musk | **0.2613±0.0325** | 0.342±0.0175 | 0.0214 |
| parkinsons | 0.1876±0.036 | 0.1814±0.0369 | 0.656 |
| pop_failures | 0.0818±0.0064 | 0.0803±0.0077 | 0.6483 |
| sonar | 0.3769±0.0249 | **0.2846±0.0474** | 0.0195 |
| spectf | 0.2466±0.0381 | 0.203±0.0255 | 0.0859 |
| vehicle | **0.4241±0.0223** | 0.522±0.0157 | 0.0018 |
| waveform | **0.1774±0.0016** | 0.2218±0.0048 | 1e-04 |
| wdbc | 0.3789±0.0101 | 0.4014±0.019 | 0.0899 |

**Table C.14:** Summary of classification error results on UCI datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1374±0.0293** | 0.1178±0.0348 | 0 |
| ma_colon_tumor | **0.1958±0.0965** | 0.1315±0.0937 | 0 |
| ma_gcm | **0.3582±0.1049** | 0.2267±0.1583 | 0 |
| ma_leukemia | **0.2542±0.1157** | 0.2128±0.0859 | 0.0425 |
| ma_lung_cancer | **0.536±0.0639** | 0.3616±0.091 | 0 |
| ma_prostate_cancer | **0.2995±0.1365** | 0.1069±0.12 | 0 |

**Table C.15:** Summary of stability results on Microarray datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3333±0.0642 | 0.3542±0.051 | 0.3739 |
| ma_colon_tumor | 0.3419±0.0669 | 0.3548±0.0395 | 0.7489 |
| ma_gcm | 0.6442±0.0424 | 0.6526±0.0197 | 0.6651 |
| ma_leukemia | 0.1389±0.034 | 0.1556±0.0913 | 0.7527 |
| ma_lung_cancer | 0.0667±0.0444 | 0.0578±0.0093 | 0.7235 |
| ma_prostate_cancer | 0.2471±0.0366 | 0.2471±0.0283 | 1 |

**Table C.16:** Summary of classification error results on Microarray datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.2209±0.0938** | 0.1076±0.0745 | 0 |
| dexter | 0.2089±0.0654 | 0.2118±0.0543 | 0.8337 |
| dorothea | **0.1237±0.0332** | 0.1058±0.0285 | 0 |
| gisette | **0.5371±0.076** | 0.5033±0.121 | 0.0105 |
| madelon | 0.2519±0.1523 | **0.694±0.1876** | 0 |

**Table C.17:** Summary of stability results on NIPS datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.292±0.0311 | 0.342±0.054 | 0.0601 |
| dexter | 0.1987±0.0556 | 0.1607±0.0144 | 0.2251 |
| dorothea | 0.0821±0.0015 | 0.0852±0.0055 | 0.3456 |
| gisette | **0.0483±0.0019** | 0.053±0.0034 | 0.034 |
| madelon | 0.2946±0.0168 | **0.1432±0.0095** | 0 |

**Table C.18:** Summary of classification error results on NIPS datasets for **SimbaMiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.4249±0.2015 | 0.3663±0.1533 | 0.1094 |
| glass | 0.2744±0.1677 | 0.2557±0.1408 | 0.5541 |
| heart_statlog | **0.3748±0.178** | 0.277±0.2032 | 0.0161 |
| ionosphere | **0.2031±0.0882** | 0.0916±0.0336 | 0 |
| landsat_train | 0.2851±0.1546 | 0.2851±0.1477 | 0.9727 |
| lsvt_voice | **0.267±0.2625** | 0.2318±0.2645 | 0 |
| mammogram | **0.115±0.0712** | 0.0685±0.0591 | 0 |
| musk | **0.1436±0.0472** | 0.1047±0.0557 | 0 |
| parkinsons | **0.363±0.1681** | 0.243±0.2157 | 0 |
| pop_failures | 0.2722±0.1649 | 0.2806±0.2492 | 0.4307 |
| sonar | **0.3107±0.1554** | 0.2899±0.1619 | 1e-04 |
| spectf | 0.1027±0.0511 | **0.1269±0.0994** | 0.0139 |
| vehicle | 0.4327±0.1166 | 0.4704±0.216 | 0.9632 |
| waveform | 0.205±0.0822 | 0.2133±0.0702 | 0.4749 |
| wdbc | **0.4362±0.2878** | 0.3769±0.32 | 4e-04 |

**Table C.19:** Summary of stability results on UCI datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2969±0.0267 | 0.2911±0.0162 | 0.7202 |
| glass | 0.4374±0.0306 | 0.4879±0.0829 | 0.1473 |
| heart_statlog | 0.4074±0.0174 | **0.3378±0.0358** | 0.0214 |
| ionosphere | **0.2126±0.0204** | 0.2549±0.0138 | 6e-04 |
| landsat_train | 0.2067±0.0028 | **0.145±0.0063** | 1e-04 |
| lsvt_voice | 0.4476±0.0411 | 0.4317±0.0468 | 0.3262 |
| mammogram | 0.4233±0.0907 | 0.4186±0.0717 | 0.9465 |
| musk | **0.3008±0.0196** | 0.342±0.0175 | 0.02 |
| parkinsons | 0.1938±0.0367 | 0.1814±0.0369 | 0.7024 |
| pop_failures | 0.0818±0.007 | 0.0803±0.0077 | 0.7717 |
| sonar | 0.2942±0.0344 | 0.2846±0.0474 | 0.743 |
| spectf | 0.2±0.0269 | 0.203±0.0255 | 0.8486 |
| vehicle | **0.4364±0.0165** | 0.522±0.0157 | 0.0027 |
| waveform | 0.2281±0.0066 | 0.2218±0.0048 | 0.1666 |
| wdbc | 0.3817±0.0192 | 0.4014±0.019 | 0.0604 |

**Table C.20:** Summary of classification error results on UCI datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1544±0.0464** | 0.1178±0.0348 | 0 |
| ma_colon_tumor | **0.1888±0.0804** | 0.1315±0.0937 | 0 |
| ma_gcm | **0.3018±0.1907** | 0.2267±0.1583 | 0 |
| ma_leukemia | 0.2078±0.099 | **0.2128±0.0859** | 0.0024 |
| ma_lung_cancer | **0.3749±0.1021** | 0.3616±0.091 | 0 |
| ma_prostate_cancer | **0.1578±0.1225** | 0.1069±0.12 | 0 |

**Table C.21:** Summary of stability results on Microarray datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3583±0.0742 | 0.3542±0.051 | 0.9193 |
| ma_colon_tumor | 0.3548±0.0395 | 0.3548±0.0395 | 1 |
| ma_gcm | 0.64±0.0336 | 0.6526±0.0197 | 0.5354 |
| ma_leukemia | 0.1667±0.052 | 0.1556±0.0913 | 0.862 |
| ma_lung_cancer | 0.0533±0.0093 | 0.0578±0.0093 | 0.4766 |
| ma_prostate_cancer | 0.2824±0.0319 | 0.2471±0.0283 | 0.1533 |

**Table C.22:** Summary of classification error results on Microarray datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.233±0.1185** | 0.1076±0.0745 | 0 |
| dexter | **0.3479±0.0683** | 0.2118±0.0543 | 0 |
| dorothea | **0.3724±0.141** | 0.1058±0.0285 | 0 |
| gisette | 0.4603±0.1506 | **0.5033±0.121** | 0 |
| madelon | 0.6119±0.1663 | **0.694±0.1876** | 0 |

**Table C.23:** Summary of stability results on NIPS datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.272±0.0455 | 0.342±0.054 | 0.0924 |
| dexter | 0.2013±0.0145 | **0.1607±0.0144** | 0.0162 |
| dorothea | 0.097±0.0067 | **0.0852±0.0055** | 0.0185 |
| gisette | 0.0694±0.0034 | **0.053±0.0034** | 0.0039 |
| madelon | **0.1357±0.0092** | 0.1432±0.0095 | 0.0219 |

**Table C.24:** Summary of classification error results on NIPS datasets for **SimbaLiw** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.4769±0.1481 | 0.3665±0.1721 | 0.2969 |
| glass | 0.2123±0.1339 | 0.2699±0.1934 | 0.25 |
| heart_statlog | 0.2945±0.1499 | 0.3421±0.1688 | 0.3983 |
| ionosphere | **0.261±0.1223** | 0.1226±0.0583 | 0 |
| landsat_train | **0.6031±0.1146** | 0.2695±0.1081 | 0 |
| lsvt_voice | 0.2079±0.1913 | **0.3099±0.2318** | 0 |
| mammogram | 0.2142±0.0982 | 0.1851±0.1233 | 0.1651 |
| musk | **0.2481±0.0672** | 0.1552±0.0666 | 0 |
| parkinsons | **0.5061±0.1601** | 0.2887±0.2483 | 0 |
| pop_failures | **0.3754±0.2649** | 0.2382±0.2328 | 9e-04 |
| sonar | 0.3962±0.1487 | 0.3842±0.1671 | 0.143 |
| spectf | **0.2545±0.1098** | 0.0471±0.0464 | 0 |
| vehicle | 0.4998±0.17 | 0.4933±0.2121 | 0.6112 |
| waveform | **0.5401±0.0927** | 0.28±0.1154 | 0 |
| wdbc | **0.7132±0.1569** | 0.3363±0.3223 | 0 |

**Table C.25:** Summary of stability results on UCI datasets for **SimbaMiw** with *order*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2557±0.0137 | 0.2677±0.0145 | 0.2423 |
| glass | 0.4262±0.0497 | 0.4879±0.053 | 0.1218 |
| heart_statlog | 0.363±0.0257 | 0.4119±0.0334 | 0.0959 |
| ionosphere | **0.176±0.0063** | 0.2114±0.0194 | 0.0112 |
| landsat_train | **0.1226±0.0039** | 0.1473±0.0063 | 0.0015 |
| lsvt_voice | 0.4508±0.0741 | 0.4254±0.0609 | 0.6895 |
| mammogram | 0.5023±0.0709 | **0.3209±0.0416** | 0.0041 |
| musk | **0.2504±0.0132** | 0.3756±0.0239 | 0.0011 |
| parkinsons | 0.2062±0.0292 | 0.2062±0.0273 | 1 |
| pop_failures | 0.0781±0.0026 | 0.0818±0.0037 | 0.189 |
| sonar | 0.3788±0.0161 | **0.3019±0.0388** | 0.0255 |
| spectf | 0.2256±0.0232 | 0.2165±0.0263 | 0.5291 |
| vehicle | **0.4241±0.0169** | 0.5177±0.0093 | 0.0011 |
| waveform | **0.1766±0.0065** | 0.2317±0.0061 | 0 |
| wdbc | 0.3866±0.0253 | 0.381±0.026 | 0.7876 |

**Table C.26:** Summary of classification error results on UCI datasets for **SimbaMiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1786±0.0387** | 0.1512±0.0332 | 0 |
| ma_colon_tumor | 0.2018±0.0787 | 0.1847±0.1074 | 0.3673 |
| ma_gcm | **0.4344±0.1018** | 0.2936±0.1835 | 0 |
| ma_leukemia | **0.2953±0.1059** | 0.1589±0.094 | 0 |
| ma_lung_cancer | **0.5407±0.0518** | 0.3622±0.086 | 0 |
| ma_prostate_cancer | **0.3821±0.1487** | 0.1161±0.1478 | 0 |

**Table C.27:** Summary of stability results on Microarray datasets for **SimbaMiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3125±0.0751 | 0.2958±0.0373 | 0.5965 |
| ma_colon_tumor | 0.3806±0.077 | 0.3871±0.0395 | 0.9001 |
| ma_gcm | 0.6337±0.049 | 0.6632±0.0387 | 0.2804 |
| ma_leukemia | 0.1278±0.0576 | 0.1222±0.0505 | 0.8276 |
| ma_lung_cancer | 0.0267±0.0127 | 0.0511±0.0405 | 0.3404 |
| ma_prostate_cancer | 0.25±0.0389 | 0.2618±0.0381 | 0.6797 |

**Table C.28:** Summary of classification error results on Microarray datasets for **SimbaMiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.3023±0.1007** | 0.1407±0.1042 | 0 |
| dexter | **0.2894±0.0852** | 0.2379±0.0622 | 0 |
| dorothea | **0.1692±0.0618** | 0.1237±0.0343 | 0 |
| gisette | **0.5947±0.0577** | 0.5108±0.1306 | 0 |
| madelon | 0.3068±0.1573 | **0.6944±0.1825** | 0 |

**Table C.29:** Summary of stability results on NIPS datasets for **SimbaMiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.284±0.0541 | 0.268±0.0148 | 0.5821 |
| dexter | **0.1827±0.0098** | 0.2173±0.0223 | 0.0334 |
| dorothea | 0.081±0.0047 | 0.0797±0.0069 | 0.7205 |
| gisette | **0.0477±0.0013** | 0.055±0.0043 | 0.0352 |
| madelon | 0.2843±0.0042 | **0.1343±0.0088** | 0 |

**Table C.30:** Summary of classification error results on NIPS datasets for **SimbaMiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.4222±0.2166 | 0.3665±0.1721 | 0.2188 |
| glass | 0.3021±0.1662 | 0.2699±0.1934 | 0.9453 |
| heart_statlog | 0.3014±0.2638 | 0.3421±0.1688 | 0.1514 |
| ionosphere | **0.2506±0.0718** | 0.1226±0.0583 | 0 |
| landsat_train | **0.5528±0.1915** | 0.2695±0.1081 | 0 |
| lsvt_voice | **0.3525±0.2803** | 0.3099±0.2318 | 0 |
| mammogram | 0.105±0.089 | **0.1851±0.1233** | 0 |
| musk | 0.1148±0.0283 | **0.1552±0.0666** | 0 |
| parkinsons | **0.3443±0.2168** | 0.2887±0.2483 | 0.0012 |
| pop_failures | 0.2394±0.1677 | 0.2382±0.2328 | 0.2243 |
| sonar | 0.3293±0.1759 | **0.3842±0.1671** | 0 |
| spectf | 0.0495±0.0366 | 0.0471±0.0464 | 0.1075 |
| vehicle | 0.3523±0.0909 | **0.4933±0.2121** | 0.011 |
| waveform | 0.0844±0.062 | **0.28±0.1154** | 0 |
| wdbc | 0.2625±0.2898 | **0.3363±0.3223** | 0 |

**Table C.31:** Summary of stability results on UCI datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2807±0.0153 | **0.2677±0.0145** | 0.026 |
| glass | 0.4673±0.0187 | 0.4879±0.053 | 0.5509 |
| heart_statlog | 0.3807±0.0238 | 0.4119±0.0334 | 0.2434 |
| ionosphere | 0.2526±0.0263 | **0.2114±0.0194** | 0.0086 |
| landsat_train | 0.1981±0.0049 | **0.1473±0.0063** | 2e-04 |
| lsvt_voice | 0.3619±0.0207 | 0.4254±0.0609 | 0.126 |
| mammogram | 0.4186±0.0678 | 0.3209±0.0416 | 0.0863 |
| musk | 0.384±0.0365 | 0.3756±0.0239 | 0.3513 |
| parkinsons | 0.2186±0.0523 | 0.2062±0.0273 | 0.6807 |
| pop_failures | 0.0796±0.0086 | 0.0818±0.0037 | 0.4263 |
| sonar | 0.3058±0.0307 | 0.3019±0.0388 | 0.9062 |
| spectf | 0.2391±0.0423 | 0.2165±0.0263 | 0.4049 |
| vehicle | **0.453±0.0308** | 0.5177±0.0093 | 0.0069 |
| waveform | 0.254±0.0074 | **0.2317±0.0061** | 0.0135 |
| wdbc | 0.3817±0.0265 | 0.381±0.026 | 0.9675 |

**Table C.32:** Summary of classification error results on UCI datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.1802±0.0274** | 0.1512±0.0332 | 0 |
| ma_colon_tumor | **0.2734±0.1049** | 0.1847±0.1074 | 0 |
| ma_gcm | **0.5075±0.1319** | 0.2936±0.1835 | 0 |
| ma_leukemia | **0.2719±0.1177** | 0.1589±0.094 | 0 |
| ma_lung_cancer | **0.4557±0.0902** | 0.3622±0.086 | 0 |
| ma_prostate_cancer | **0.1866±0.1146** | 0.1161±0.1478 | 0 |

**Table C.33:** Summary of stability results on Microarray datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.2583±0.0316 | 0.2958±0.0373 | 0.121 |
| ma_colon_tumor | 0.3226±0.0684 | 0.3871±0.0395 | 0.1161 |
| ma_gcm | 0.7179±0.0292 | **0.6632±0.0387** | 0.0237 |
| ma_leukemia | 0.1444±0.0663 | 0.1222±0.0505 | 0.6135 |
| ma_lung_cancer | 0.0556±0.0136 | 0.0511±0.0405 | 0.862 |
| ma_prostate_cancer | 0.2206±0.0403 | 0.2618±0.0381 | 0.2312 |

**Table C.34:** Summary of classification error results on Microarray datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.2404±0.0938** | 0.1407±0.1042 | 0 |
| dexter | **0.3185±0.0622** | 0.2379±0.0622 | 0 |
| dorothea | **0.2275±0.0827** | 0.1237±0.0343 | 0 |
| gisette | **0.5558±0.1249** | 0.5108±0.1306 | 0 |
| madelon | 0.5895±0.1594 | **0.6944±0.1825** | 0 |

**Table C.35:** Summary of stability results on NIPS datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.358±0.0572 | **0.268±0.0148** | 0.0231 |
| dexter | 0.2113±0.0107 | 0.2173±0.0223 | 0.6959 |
| dorothea | 0.0877±0.0038 | 0.0797±0.0069 | 0.0949 |
| gisette | 0.0512±0.0027 | 0.055±0.0043 | 0.2234 |
| madelon | 0.1568±0.0056 | **0.1343±0.0088** | 0.0176 |

**Table C.36:** Summary of classification error results on NIPS datasets for **SimbaLiw** with $order\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | **0.0239±0.0621** | -0.0209±0.0466 | 0.0156 |
| heart_statlog | 0.0076±0.0368 | -0.0248±0.0363 | 0.064 |
| ionosphere | -0.0092±0.0319 | -0.0195±0.0354 | 0.3263 |
| landsat_train | -0.021±0.0204 | -0.0158±0.031 | 0.2726 |
| leaf | -0.0572±0.0214 | -0.0492±0.038 | 0.5417 |
| lsvt_voice | 0.036±0.0532 | 0.0348±0.0584 | 0.3442 |
| mammogram | 0.0248±0.0227 | 0.0157±0.0447 | 0.3654 |
| musk | -0.0216±0.0204 | **-0.0139±0.022** | 0 |
| parkinsons | **0.0703±0.0448** | 0.0305±0.0309 | 0.0014 |
| pop_failures | -0.0619±0.0269 | -0.0527±0.017 | 0.2243 |
| sonar | -0.0134±0.0134 | **-0.0074±0.0183** | 0.0178 |
| spectf | -0.0024±0.0304 | 0.0056±0.0169 | 0.2022 |
| vehicle | **0.0308±0.0538** | 0.0064±0.0871 | 0.0395 |
| waveform | -0.0415±0.019 | -0.0402±0.0342 | 0.8695 |
| wdbc | -0.0308±0.024 | **6e-04±0.0212** | 1e-04 |

**Table C.37:** Summary of stability results on UCI datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | **0.2568±0.0091** | 0.2911±0.0131 | 0.0158 |
| heart_statlog | 0.3704±0.0343 | 0.3852±0.0474 | 0.6453 |
| ionosphere | **0.1726±0.0117** | 0.2366±0.0287 | 0.007 |
| landsat_train | 0.1274±0.0028 | 0.1318±0.0079 | 0.2937 |
| leaf | 0.8353±0.015 | **0.7341±0.0261** | 0.001 |
| lsvt_voice | 0.4571±0.0362 | 0.4222±0.0774 | 0.295 |
| mammogram | 0.5535±0.0761 | 0.5395±0.0861 | 0.8047 |
| musk | 0.3908±0.0181 | **0.316±0.0202** | 9e-04 |
| parkinsons | 0.1691±0.0297 | 0.1856±0.0292 | 0.3653 |
| pop_failures | 0.0877±0.002 | 0.0892±0.0037 | 0.4766 |
| sonar | 0.4154±0.0583 | 0.375±0.0296 | 0.3467 |
| spectf | 0.2511±0.0351 | 0.2511±0.0355 | 1 |
| vehicle | **0.4255±0.0124** | 0.5324±0.0228 | 6e-04 |
| waveform | **0.223±0.007** | 0.3011±0.0117 | 4e-04 |
| wdbc | **0.2923±0.0238** | 0.3965±0.0214 | 0.002 |

**Table C.38:** Summary of classification error results on UCI datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | -0.0316±0.0083 | **-0.013±0.0069** | 0 |
| ma_colon_tumor | **-0.0018±0.014** | -0.0206±0.0199 | 0 |
| ma_gcm | -0.0108±0.0149 | **0.012±0.0114** | 0 |
| ma_leukemia | -0.0112±0.0109 | **-0.0078±0.0145** | 0.0385 |
| ma_lung_cancer | -0.0215±0.0138 | **-0.0147±0.0075** | 0 |
| ma_prostate_cancer | -0.0024±0.0302 | -0.0041±0.0169 | 0.6472 |

**Table C.39:** Summary of stability results on Microarray datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3417±0.0745 | 0.2708±0.0675 | 0.2611 |
| ma_colon_tumor | 0.271±0.0629 | 0.2516±0.077 | 0.6072 |
| ma_gcm | 0.6589±0.0205 | 0.6295±0.0485 | 0.166 |
| ma_leukemia | 0.1111±0.0393 | 0.0667±0.0465 | 0.1778 |
| ma_lung_cancer | 0.0644±0.0165 | 0.08±0.0093 | 0.1836 |
| ma_prostate_cancer | **0.2765±0.0619** | 0.3676±0.0375 | 0.0143 |

**Table C.40:** Summary of classification error results on Microarray datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **-0.0118±0.0231** | -0.0164±0.0158 | 0.0284 |
| dexter | **0.1208±0.0887** | 0.0883±0.0788 | 0 |
| dorothea | 0.1436±0.0975 | **0.2994±0.1185** | 0 |
| gisette | **-0.0059±0.0101** | -0.007±0.0159 | 0.0211 |
| madelon | -0.0321±0.0081 | **-0.0159±0.0068** | 0 |

**Table C.41:** Summary of stability results on NIPS datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.274±0.0573 | 0.322±0.0415 | 0.2151 |
| dexter | 0.2793±0.0511 | 0.2413±0.0141 | 0.1989 |
| dorothea | 0.0824±0.0029 | 0.0762±0.0087 | 0.2236 |
| gisette | **0.0883±0.0047** | 0.1194±0.0035 | 7e-04 |
| madelon | 0.4963±0.0108 | **0.4085±0.0058** | 2e-04 |

**Table C.42:** Summary of classification error results on NIPS datasets for **MBIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | -0.044±0.0391 | -0.0209±0.0466 | 0.5294 |
| heart_statlog | **0.0337±0.0347** | -0.0248±0.0363 | 0.0049 |
| ionosphere | **-0.0035±0.0302** | -0.0195±0.0354 | 0.0197 |
| landsat_train | -0.0013±0.0256 | -0.0158±0.031 | 0.0515 |
| leaf | -0.0489±0.0341 | -0.0492±0.038 | 0.4561 |
| lsvt_voice | 0.0078±0.0493 | **0.0348±0.0584** | 0 |
| mammogram | -0.0337±0.017 | **0.0157±0.0447** | 0 |
| musk | **0.0154±0.0137** | -0.0139±0.022 | 0 |
| parkinsons | -0.0095±0.0237 | **0.0305±0.0309** | 0 |
| pop_failures | **-0.042±0.0168** | -0.0527±0.017 | 0.0382 |
| sonar | **-0.0019±0.0178** | -0.0074±0.0183 | 0.0339 |
| spectf | 0.0074±0.025 | 0.0056±0.0169 | 0.5105 |
| vehicle | 0.0091±0.052 | 0.0064±0.0871 | 0.4851 |
| waveform | -0.0339±0.0222 | -0.0402±0.0342 | 0.4813 |
| wdbc | **0.0771±0.0582** | 6e-04±0.0212 | 0 |

**Table C.43:** Summary of stability results on UCI datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | **0.2542±0.0264** | 0.2911±0.0131 | 0.0114 |
| heart_statlog | 0.3481±0.0181 | 0.3852±0.0474 | 0.1634 |
| ionosphere | 0.1829±0.0259 | 0.2366±0.0287 | 0.0789 |
| landsat_train | 0.1561±0.0055 | **0.1318±0.0079** | 0.0016 |
| leaf | 0.7435±0.0206 | 0.7341±0.0261 | 0.3375 |
| lsvt_voice | 0.4476±0.0566 | 0.4222±0.0774 | 0.6556 |
| mammogram | 0.4791±0.0535 | 0.5395±0.0861 | 0.3203 |
| musk | **0.2916±0.0265** | 0.316±0.0202 | 0.0307 |
| parkinsons | 0.1753±0.0146 | 0.1856±0.0292 | 0.5185 |
| pop_failures | 0.0922±0.0072 | 0.0892±0.0037 | 0.405 |
| sonar | 0.3942±0.0414 | 0.375±0.0296 | 0.5083 |
| spectf | 0.2662±0.033 | 0.2511±0.0355 | 0.5393 |
| vehicle | 0.5139±0.0169 | 0.5324±0.0228 | 0.2347 |
| waveform | 0.3499±0.0051 | **0.3011±0.0117** | 5e-04 |
| wdbc | **0.0944±0.0164** | 0.3965±0.0214 | 0 |

**Table C.44:** Summary of classification error results on UCI datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | -0.0193±0.0073 | **-0.013±0.0069** | 0 |
| ma_colon_tumor | **-0.0143±0.0155** | -0.0206±0.0199 | 0.0333 |
| ma_gcm | 0.0021±0.0127 | **0.012±0.0114** | 0 |
| ma_leukemia | -0.0108±0.011 | **-0.0078±0.0145** | 0.02 |
| ma_lung_cancer | -0.0263±0.0106 | **-0.0147±0.0075** | 0 |
| ma_prostate_cancer | -0.0139±0.0135 | **-0.0041±0.0169** | 0 |

**Table C.45:** Summary of stability results on Microarray datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.2875±0.0427 | 0.2708±0.0675 | 0.7235 |
| ma_colon_tumor | 0.3871±0.0323 | **0.2516±0.077** | 0.0146 |
| ma_gcm | 0.6695±0.0191 | 0.6295±0.0485 | 0.1786 |
| ma_leukemia | 0.1222±0.0954 | 0.0667±0.0465 | 0.089 |
| ma_lung_cancer | 0.0644±0.0093 | 0.08±0.0093 | 0.1079 |
| ma_prostate_cancer | **0.2294±0.0132** | 0.3676±0.0375 | 0.0033 |

**Table C.46:** Summary of classification error results on Microarray datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.0252±0.0218** | -0.0164±0.0158 | 0 |
| dexter | **0.1548±0.0996** | 0.0883±0.0788 | 0 |
| dorothea | **0.4635±0.2009** | 0.2994±0.1185 | 0 |
| gisette | -0.0172±0.0082 | **-0.007±0.0159** | 0 |
| madelon | -0.0204±0.0058 | **-0.0159±0.0068** | 0 |

**Table C.47:** Summary of stability results on NIPS datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.34±0.0367 | 0.322±0.0415 | 0.6152 |
| dexter | 0.2207±0.0261 | 0.2413±0.0141 | 0.3133 |
| dorothea | 0.0897±0.0044 | **0.0762±0.0087** | 0.0415 |
| gisette | **0.0698±0.0035** | 0.1194±0.0035 | 0 |
| madelon | **0.3902±0.0109** | 0.4085±0.0058 | 0.0358 |

**Table C.48:** Summary of classification error results on NIPS datasets for **RLIW + Relief** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | -0.013±0.0626 | -0.0232±0.0419 | 1 |
| heart_statlog | **-0.0268±0.029** | -0.0644±0.0225 | 0.0068 |
| ionosphere | -0.0259±0.0261 | -0.0183±0.0245 | 0.0602 |
| landsat_train | 0.0075±0.0169 | 7e-04±0.0317 | 0.1609 |
| leaf | 0.0176±0.0306 | -0.0259±0.0689 | 0.0803 |
| lsvt_voice | -0.0135±0.0107 | **-0.0087±0.0118** | 0 |
| mammogram | **0.0113±0.0267** | -0.0259±0.0192 | 0 |
| musk | **-0.006±0.0086** | -0.0292±0.0079 | 0 |
| parkinsons | 0.0677±0.0602 | **0.1276±0.0869** | 0.0029 |
| pop_failures | -0.0642±0.0236 | **0.0024±0.0275** | 2e-04 |
| sonar | **-0.0044±0.0183** | -0.0317±0.0127 | 0 |
| spectf | -0.0213±0.0179 | **0.0147±0.0339** | 0 |
| vehicle | **0.0429±0.0403** | -0.0246±0.0287 | 0 |
| waveform | -0.0516±0.0216 | -0.038±0.0213 | 0.08 |
| wdbc | -0.0304±0.0316 | **0.0329±0.0391** | 0 |

**Table C.49:** Summary of stability results on UCI datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.3432±0.0203 | 0.3521±0.0151 | 0.4591 |
| heart_statlog | 0.3289±0.0525 | 0.3481±0.0367 | 0.6493 |
| ionosphere | 0.1726±0.0153 | 0.192±0.0244 | 0.3073 |
| landsat_train | 0.1252±0.0047 | 0.1294±0.0024 | 0.1847 |
| leaf | **0.6671±0.0411** | 0.8647±0.0208 | 0.0015 |
| lsvt_voice | 0.4825±0.0611 | **0.4159±0.0174** | 0.0393 |
| mammogram | 0.4512±0.0606 | 0.5023±0.0265 | 0.1894 |
| musk | 0.2756±0.0215 | 0.2748±0.0198 | 0.9623 |
| parkinsons | 0.1897±0.0237 | 0.1608±0.0156 | 0.1148 |
| pop_failures | 0.0862±0.0017 | 0.0967±0.0083 | 0.08 |
| sonar | 0.4308±0.0691 | **0.2385±0.0356** | 3e-04 |
| spectf | 0.2436±0.0181 | 0.2301±0.0351 | 0.2552 |
| vehicle | **0.4014±0.0203** | 0.6043±0.0477 | 0.0015 |
| waveform | 0.5371±0.0108 | **0.2426±0.0095** | 0 |
| wdbc | 0.1035±0.0107 | **0.0775±0.007** | 0.0164 |

**Table C.50:** Summary of classification error results on UCI datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | -0.0154±0.0066 | -0.0164±0.0084 | 0.4003 |
| ma_colon_tumor | **-0.0152±0.0128** | -0.0219±0.0065 | 0 |
| ma_gcm | **-0.0106±0.0107** | -0.0149±0.0137 | 2e-04 |
| ma_leukemia | -0.0263±0.0077 | **-0.0212±0.0093** | 0 |
| ma_lung_cancer | -0.0287±0.0071 | **-0.0099±0.0086** | 0 |
| ma_prostate_cancer | -0.0135±0.0089 | **0.0011±0.0117** | 0 |

**Table C.51:** Summary of stability results on Microarray datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.2083±0.0295** | 0.3583±0.1014 | 0.0387 |
| ma_colon_tumor | 0.3226±0.0456 | 0.3742±0.0743 | 0.3058 |
| ma_gcm | 0.6674±0.0377 | 0.6695±0.0284 | 0.9405 |
| ma_leukemia | 0.2056±0.0317 | **0.0611±0.0124** | 4e-04 |
| ma_lung_cancer | 0.0756±0.0199 | 0.0689±0.005 | 0.4676 |
| ma_prostate_cancer | 0.3118±0.0218 | 0.2853±0.0305 | 0.2761 |

**Table C.52:** Summary of classification error results on Microarray datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | -0.0181±0.008 | **-0.0102±0.0149** | 0 |
| dexter | 0.002±0.0179 | **0.0192±0.0338** | 0 |
| dorothea | **0.3164±0.1742** | 0.1915±0.0978 | 0 |
| gisette | 1e-04±0.0147 | 0.001±0.0082 | 0.1011 |
| madelon | -0.0293±0.0052 | **-0.0194±0.0073** | 0 |

**Table C.53:** Summary of stability results on NIPS datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.288±0.0517 | 0.362±0.0164 | 0.0596 |
| dexter | **0.196±0.0055** | 0.3107±0.0269 | 6e-04 |
| dorothea | 0.0877±0.0047 | 0.0863±0.0083 | 0.5275 |
| gisette | **0.0718±0.0019** | 0.0788±0.0018 | 0.0049 |
| madelon | **0.3803±0.0081** | 0.4294±0.0124 | 0.0051 |

**Table C.54:** Summary of classification error results on NIPS datasets for **MBIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.0348±0.0779 | -0.0232±0.0419 | 0.0781 |
| heart_statlog | **0.0052±0.0508** | -0.0644±0.0225 | 0.0068 |
| ionosphere | -0.0462±0.024 | **-0.0183±0.0245** | 2e-04 |
| landsat_train | -0.0301±0.0186 | **7e-04±0.0317** | 1e-04 |
| leaf | -0.0276±0.0415 | -0.0259±0.0689 | 1 |
| lsvt_voice | -0.0242±0.0084 | **-0.0087±0.0118** | 0 |
| mammogram | **-0.0075±0.017** | -0.0259±0.0192 | 0 |
| musk | **-0.0226±0.0146** | -0.0292±0.0079 | 0 |
| parkinsons | 0.0984±0.0944 | 0.1276±0.0869 | 0.2877 |
| pop_failures | -0.0324±0.025 | **0.0024±0.0275** | 0.0056 |
| sonar | **-0.0186±0.0175** | -0.0317±0.0127 | 1e-04 |
| spectf | **0.0343±0.0354** | 0.0147±0.0339 | 1e-04 |
| vehicle | -0.0146±0.0329 | -0.0246±0.0287 | 0.4586 |
| waveform | **-0.0208±0.0263** | -0.038±0.0213 | 0.0296 |
| wdbc | 0.0223±0.0305 | **0.0329±0.0391** | 0.0071 |

**Table C.55:** Summary of stability results on UCI datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | **0.2688±0.0302** | 0.3521±0.0151 | 0.0044 |
| heart_statlog | 0.3481±0.0228 | 0.3481±0.0367 | 1 |
| ionosphere | 0.1863±0.0244 | 0.192±0.0244 | 0.6584 |
| landsat_train | 0.1272±0.0052 | 0.1294±0.0024 | 0.4836 |
| leaf | **0.7765±0.0093** | 0.8647±0.0208 | 0.001 |
| lsvt_voice | 0.5111±0.0133 | **0.4159±0.0174** | 7e-04 |
| mammogram | 0.4837±0.0861 | 0.5023±0.0265 | 0.6702 |
| musk | 0.2664±0.0278 | 0.2748±0.0198 | 0.6557 |
| parkinsons | 0.2103±0.0347 | **0.1608±0.0156** | 0.0078 |
| pop_failures | **0.0773±0.0085** | 0.0967±0.0083 | 0.0029 |
| sonar | 0.3904±0.0323 | **0.2385±0.0356** | 5e-04 |
| spectf | 0.2346±0.0312 | 0.2301±0.0351 | 0.8733 |
| vehicle | **0.4416±0.0133** | 0.6043±0.0477 | 0.0026 |
| waveform | **0.1972±0.0052** | 0.2426±0.0095 | 0.0014 |
| wdbc | 0.0845±0.0114 | 0.0775±0.007 | 0.3894 |

**Table C.56:** Summary of classification error results on UCI datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | -0.0154±0.0085 | -0.0164±0.0084 | 0.2333 |
| ma_colon_tumor | **-0.0183±0.0079** | -0.0219±0.0065 | 1e-04 |
| ma_gcm | **-0.0113±0.0113** | -0.0149±0.0137 | 0 |
| ma_leukemia | -0.0188±0.0072 | -0.0212±0.0093 | 0.1116 |
| ma_lung_cancer | -0.0162±0.0064 | **-0.0099±0.0086** | 0 |
| ma_prostate_cancer | -0.0182±0.0062 | **0.0011±0.0117** | 0 |

**Table C.57:** Summary of stability results on Microarray datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.4125±0.0913 | 0.3583±0.1014 | 0.5206 |
| ma_colon_tumor | 0.3097±0.054 | 0.3742±0.0743 | 0.2488 |
| ma_gcm | 0.6589±0.0362 | 0.6695±0.0284 | 0.635 |
| ma_leukemia | 0.2667±0.1465 | **0.0611±0.0124** | 0.0378 |
| ma_lung_cancer | 0.0667±0.0136 | 0.0689±0.005 | 0.778 |
| ma_prostate_cancer | 0.2588±0.0322 | 0.2853±0.0305 | 0.2552 |

**Table C.58:** Summary of classification error results on Microarray datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | -0.0192±0.0094 | **-0.0102±0.0149** | 0 |
| dexter | **0.0913±0.0494** | 0.0192±0.0338 | 0 |
| dorothea | **0.565±0.2223** | 0.1915±0.0978 | 0 |
| gisette | **0.0182±0.0265** | 0.001±0.0082 | 0 |
| madelon | -0.0249±0.0056 | **-0.0194±0.0073** | 0 |

**Table C.59:** Summary of stability results on NIPS datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | **0.258±0.0319** | 0.362±0.0164 | 0.0018 |
| dexter | **0.2153±0.0234** | 0.3107±0.0269 | 0.0031 |
| dorothea | 0.0863±0.0026 | 0.0863±0.0083 | 1 |
| gisette | **0.0573±0.0014** | 0.0788±0.0018 | 0 |
| madelon | **0.3486±0.0116** | 0.4294±0.0124 | 3e-04 |

**Table C.60:** Summary of classification error results on NIPS datasets for **RLIW + RandomForests** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.9778±0 | 0.9778±0 | |
| heart_statlog | 0.9778±0 | 0.9366±0.0696 | 0.1003 |
| ionosphere | **0.9778±0** | 0.9497±0.0399 | 1e-04 |
| leaf | 0.9778±0 | 0.9778±0 | |
| mammogram | **0.9548±0.0796** | 0.9328±0.1062 | 3e-04 |
| musk | 0.9715±0.0193 | **0.9778±0** | 0 |
| parkinsons | 0.9272±0.044 | **0.9778±0** | 2e-04 |
| pop_failures | 0.9778±0 | 0.9778±0 | |
| sonar | **0.9582±0.0605** | 0.8885±0.1151 | 0 |
| spectf | 0.9373±0.0423 | **0.9778±0** | 0 |
| vehicle | 0.9778±0 | 0.9778±0 | |
| wdbc | 0.9054±0.0712 | **0.9778±0** | 0 |

**Table C.61:** Summary of stability results on UCI datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.349±0 | ltNA 0.349±0 | |
| heart_statlog | 0.4±0 | ltNA 0.4±0 | |
| ionosphere | 0.3486±0 | ltNA 0.3486±0 | |
| leaf | 0.9647±0.0093 | 0.9635±0.0134 | 0.9023 |
| mammogram | 0.6047±0.0285 | 0.5349±0.087 | 0.1841 |
| musk | 0.4244±0 | ltNA 0.4244±0 | |
| parkinsons | 0.2165±0 | ltNA 0.2165±0 | |
| pop_failures | 0.0855±0 | ltNA 0.0855±0 | |
| sonar | 0.5231±0.0211 | 0.5154±0.0161 | 0.405 |
| spectf | 0.1955±0 | ltNA 0.1955±0 | |
| vehicle | 0.7589±0.0082 | 0.7556±0.0027 | 0.3846 |
| wdbc | 0.3873±0 | ltNA 0.3873±0 | |

**Table C.62:** Summary of classification error results on UCI datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.9576±0.0468 | **0.9708±0.0173** | 0 |
| ma_colon_tumor | 0.9091±0.1059 | **0.9194±0.1291** | 0 |
| ma_gcm | **0.9778±0** | 0.9748±0.0176 | 0.0059 |
| ma_leukemia | 0.9539±0.0528 | 0.9504±0.0742 | 0.9247 |
| ma_lung_cancer | **0.9709±0.0173** | 0.9406±0.0885 | 0 |
| ma_prostate_cancer | 0.9712±0.0174 | **0.9778±0** | 0 |

**Table C.63:** Summary of stability results on Microarray datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.575±0.0186 | 0.5833±0.051 | 0.7489 |
| ma_colon_tumor | **0.3548±0.1094** | 0.5226±0.0803 | 0.0109 |
| ma_gcm | 0.8758±0.0485 | 0.8842±0.0494 | 0.8297 |
| ma_leukemia | 0.3333±0 | ltNA 0.3333±0 | |
| ma_lung_cancer | 0.1889±0 | **0.0956±0.0169** | 2e-04 |
| ma_prostate_cancer | 0.5412±0.0677 | 0.5382±0.0638 | 0.941 |

**Table C.64:** Summary of classification error results on Microarray datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.9695±0.0224 | **0.9778±0** | 0 |
| dexter | 0.9778±0 | 0.9778±0 | |
| dorothea | 0.9778±0 | 0.9778±0 | |
| madelon | 0.9778±0 | 0.9778±0 | |

**Table C.65:** Summary of stability results on NIPS datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.48±0.0728 | 0.462±0.0716 | 0.3739 |
| dexter | 0.5293±0.0322 | 0.5373±0.0348 | 0.4144 |
| dorothea | 0.1009±0 | ltNA 0.1009±0 | |
| madelon | 0.5035±0.0128 | 0.5049±0.0099 | 0.8968 |

**Table C.66:** Summary of classification error results on NIPS datasets for **MBIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.8623±0.0673 | **0.9778±0** | 0.035 |
| heart_statlog | 0.9778±0 | 0.9366±0.0696 | 0.1003 |
| ionosphere | 0.9519±0.0407 | 0.9497±0.0399 | 0.0906 |
| leaf | 0.9778±0 | 0.9778±0 | |
| mammogram | **0.9778±0** | 0.9328±0.1062 | 2e-04 |
| musk | 0.9694±0.0188 | **0.9778±0** | 0 |
| parkinsons | 0.9368±0.0484 | **0.9778±0** | 0.0011 |
| pop_failures | 0.9778±0 | 0.9778±0 | |
| sonar | **0.9614±0.0456** | 0.8885±0.1151 | 0 |
| spectf | 0.9778±0 | 0.9778±0 | |
| vehicle | 0.9778±0 | 0.9778±0 | |
| wdbc | 0.9778±0 | 0.9778±0 | |

**Table C.67:** Summary of stability results on UCI datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.349±0 | ltNA 0.349±0 | |
| heart_statlog | 0.4±0 | ltNA 0.4±0 | |
| ionosphere | 0.3486±0 | ltNA 0.3486±0 | |
| leaf | 0.9671±0.0159 | 0.9635±0.0134 | 0.7407 |
| mammogram | 0.5256±0.0816 | 0.5349±0.087 | 0.896 |
| musk | 0.4244±0 | ltNA 0.4244±0 | |
| parkinsons | 0.2165±0 | ltNA 0.2165±0 | |
| pop_failures | 0.0855±0 | ltNA 0.0855±0 | |
| sonar | 0.5462±0.0387 | 0.5154±0.0161 | 0.1733 |
| spectf | 0.1955±0 | ltNA 0.1955±0 | |
| vehicle | 0.7678±0.0114 | 0.7556±0.0027 | 0.1078 |
| wdbc | 0.3873±0 | ltNA 0.3873±0 | |

**Table C.68:** Summary of classification error results on UCI datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.9557±0.0286 | **0.9708±0.0173** | 0 |
| ma_colon_tumor | **0.9402±0.07** | 0.9194±0.1291 | 0 |
| ma_gcm | **0.9778±0** | 0.9748±0.0176 | 0.0059 |
| ma_leukemia | 0.9484±0.0693 | **0.9504±0.0742** | 0 |
| ma_lung_cancer | **0.9719±0.0176** | 0.9406±0.0885 | 0 |
| ma_prostate_cancer | 0.9778±0 | 0.9778±0 | |

**Table C.69:** Summary of stability results on Microarray datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.4±0.0401** | 0.5833±0.051 | 5e-04 |
| ma_colon_tumor | **0.2903±0.094** | 0.5226±0.0803 | 0.0137 |
| ma_gcm | 0.8632±0.0471 | 0.8842±0.0494 | 0.4565 |
| ma_leukemia | 0.3333±0 | ltNA 0.3333±0 | |
| ma_lung_cancer | 0.1889±0 | **0.0956±0.0169** | 2e-04 |
| ma_prostate_cancer | 0.5059±0.0483 | 0.5382±0.0638 | 0.3455 |

**Table C.70:** Summary of classification error results on Microarray datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.9778±0 | 0.9778±0 | |
| dexter | 0.9778±0 | 0.9778±0 | |
| dorothea | 0.9778±0 | 0.9778±0 | |
| madelon | 0.9778±0 | 0.9778±0 | |

**Table C.71:** Summary of stability results on NIPS datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.43±0 | 0.462±0.0716 | 0.3739 |
| dexter | 0.5187±0.0292 | 0.5373±0.0348 | 0.495 |
| dorothea | 0.1009±0 | ltNA 0.1009±0 | |
| madelon | 0.5042±0.0153 | 0.5049±0.0099 | 0.9025 |

**Table C.72:** Summary of classification error results on NIPS datasets for **RLIW + IG** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.9778±0 | 0.9778±0 | |
| heart_statlog | 0.9778±0 | 0.9778±0 | |
| ionosphere | **0.9778±0** | 0.9563±0.042 | 0.0025 |
| landsat_train | 0.9778±0 | 0.9778±0 | |
| leaf | 0.9778±0 | 0.9778±0 | |
| lsvt_voice | **0.9558±0.0437** | 0.9305±0.0793 | 0 |
| mammogram | **0.9566±0.028** | 0.8145±0.106 | 0 |
| musk | **0.97±0.019** | 0.9057±0.1114 | 0 |
| parkinsons | **0.9778±0** | 0.9016±0.0628 | 5e-04 |
| pop_failures | **0.9778±0** | 0.9306±0.0537 | 0.0038 |
| sonar | 0.9183±0.0658 | **0.9341±0.0454** | 0 |
| vehicle | 0.9778±0 | 0.9778±0 | |
| waveform | 0.9778±0 | 0.9778±0 | |
| wdbc | 0.8976±0.1081 | **0.9778±0** | 0 |

**Table C.73:** Summary of stability results on UCI datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2578±0.0196 | 0.2578±0.018 | 1 |
| heart_statlog | 0.357±0.0376 | 0.3881±0.038 | 0.3751 |
| ionosphere | 0.1977±0.0367 | 0.16±0.0134 | 0.0756 |
| landsat_train | 0.1256±0.0042 | 0.1241±0.0063 | 0.7391 |
| leaf | 0.6859±0.0392 | 0.6918±0.0588 | 0.8872 |
| lsvt_voice | 0.4349±0.0414 | 0.4254±0.0284 | 0.6657 |
| mammogram | 0.414±0.0666 | 0.4279±0.0208 | 0.7102 |
| musk | 0.2487±0.01 | 0.2345±0.0237 | 0.3098 |
| parkinsons | 0.1897±0.0056 | 0.1979±0.0321 | 0.5543 |
| pop_failures | 0.0781±0.0079 | 0.0751±0.0031 | 0.2943 |
| sonar | 0.3769±0.0393 | 0.3538±0.0506 | 0.4175 |
| vehicle | 0.4099±0.0195 | 0.4307±0.026 | 0.2367 |
| waveform | 0.1714±0.0061 | 0.1754±0.0047 | 0.4215 |
| wdbc | 0.3718±0.0313 | 0.3514±0.0298 | 0.3023 |

**Table C.74:** Summary of classification error results on UCI datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.9629±0.0401** | 0.9462±0.0651 | 0 |
| ma_colon_tumor | **0.9034±0.1022** | 0.8597±0.1546 | 0 |
| ma_gcm | 0.966±0.0234 | **0.9778±0** | 0 |
| ma_leukemia | 0.9222±0.0836 | **0.9581±0.0355** | 0 |
| ma_lung_cancer | **0.9695±0.017** | 0.9486±0.0538 | 0 |
| ma_prostate_cancer | **0.9759±0.0166** | 0.9706±0.0172 | 0 |

**Table C.75:** Summary of stability results on Microarray datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3333±0.0691 | 0.275±0.0309 | 0.2212 |
| ma_colon_tumor | 0.4±0.0984 | 0.3161±0.1005 | 0.2857 |
| ma_gcm | 0.6505±0.0437 | 0.6421±0.0387 | 0.7931 |
| ma_leukemia | 0.1167±0.0362 | 0.1278±0.0465 | 0.6702 |
| ma_lung_cancer | 0.0822±0.0243 | 0.0667±0.0079 | 0.2056 |
| ma_prostate_cancer | 0.3059±0.0242 | 0.2735±0.0246 | 0.0628 |

**Table C.76:** Summary of classification error results on Microarray datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.9778±0 | 0.9778±0 | |
| dexter | 0.9678±0.0287 | **0.9738±0.0124** | 0 |
| dorothea | 0.9778±0 | 0.9778±0 | |
| gisette | 0.9778±0 | 0.9778±0 | |
| madelon | 0.9778±0 | 0.9778±0 | |

**Table C.77:** Summary of stability results on NIPS datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.256±0.0602 | 0.258±0.055 | 0.9458 |
| dexter | 0.2093±0.0252 | 0.2133±0.0422 | 0.8682 |
| dorothea | 0.0821±0.0061 | 0.0828±0.0026 | 0.8446 |
| gisette | 0.0485±0.0015 | 0.0483±0.0016 | 0.8815 |
| madelon | 0.2911±0.0102 | 0.2892±0.0124 | 0.656 |

**Table C.78:** Summary of classification error results on NIPS datasets for **MBIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.9778±0 | 0.9778±0 | |
| heart_statlog | 0.9778±0 | 0.9778±0 | |
| ionosphere | **0.9778±0** | 0.9563±0.042 | 0.0025 |
| landsat_train | 0.9778±0 | 0.9778±0 | |
| leaf | 0.9778±0 | 0.9778±0 | |
| lsvt_voice | **0.9468±0.0474** | 0.9305±0.0793 | 0 |
| mammogram | **0.9778±0** | 0.8145±0.106 | 0 |
| musk | **0.9778±0** | 0.9057±0.1114 | 0 |
| parkinsons | **0.9778±0** | 0.9016±0.0628 | 5e-04 |
| pop_failures | 0.9277±0.0523 | 0.9306±0.0537 | 0.4677 |
| sonar | **0.93±0.0954** | 0.9341±0.0454 | 8e-04 |
| vehicle | 0.9778±0 | 0.9778±0 | |
| waveform | 0.9778±0 | 0.9778±0 | |
| wdbc | 0.9778±0 | 0.9778±0 | |

**Table C.79:** Summary of stability results on UCI datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| diabetes | 0.2599±0.0087 | 0.2578±0.018 | 0.7825 |
| heart_statlog | 0.3689±0.0707 | 0.3881±0.038 | 0.6177 |
| ionosphere | 0.1749±0.0125 | 0.16±0.0134 | 0.1443 |
| landsat_train | 0.1245±0.0031 | 0.1241±0.0063 | 0.9073 |
| leaf | 0.6682±0.0429 | 0.6918±0.0588 | 0.5562 |
| lsvt_voice | 0.4921±0.0251 | **0.4254±0.0284** | 0.0171 |
| mammogram | 0.4791±0.0453 | 0.4279±0.0208 | 0.0858 |
| musk | 0.258±0.0182 | 0.2345±0.0237 | 0.1522 |
| parkinsons | 0.2103±0.0237 | 0.1979±0.0321 | 0.4263 |
| pop_failures | 0.0751±0.0031 | 0.0751±0.0031 | 1 |
| sonar | 0.3538±0.0515 | 0.3538±0.0506 | 1 |
| vehicle | 0.4284±0.0139 | 0.4307±0.026 | 0.882 |
| waveform | 0.1745±0.0048 | 0.1754±0.0047 | 0.7871 |
| wdbc | 0.3465±0.0175 | 0.3514±0.0298 | 0.7882 |

**Table C.80:** Summary of classification error results on UCI datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | **0.9778±0** | 0.9462±0.0651 | 0 |
| ma_colon_tumor | **0.9357±0.0681** | 0.8597±0.1546 | 0 |
| ma_gcm | 0.9778±0 | 0.9778±0 | |
| ma_leukemia | 0.9458±0.0443 | **0.9581±0.0355** | 0 |
| ma_lung_cancer | 0.9429±0.0552 | **0.9486±0.0538** | 0 |
| ma_prostate_cancer | **0.9778±0** | 0.9706±0.0172 | 0 |

**Table C.81:** Summary of stability results on Microarray datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| ma_breast_cancer | 0.3333±0.0607 | 0.275±0.0309 | 0.1281 |
| ma_colon_tumor | 0.3161±0.062 | 0.3161±0.1005 | 1 |
| ma_gcm | 0.6716±0.041 | 0.6421±0.0387 | 0.3351 |
| ma_leukemia | 0.1167±0.0692 | 0.1278±0.0465 | 0.587 |
| ma_lung_cancer | 0.08±0.0122 | 0.0667±0.0079 | 0.1447 |
| ma_prostate_cancer | 0.3±0.0717 | 0.2735±0.0246 | 0.4103 |

**Table C.82:** Summary of classification error results on Microarray datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.9778±0 | 0.9778±0 | |
| dexter | **0.9778±0** | 0.9738±0.0124 | 0 |
| dorothea | 0.9778±0 | 0.9778±0 | |
| gisette | 0.9778±0 | 0.9778±0 | |
| madelon | 0.9778±0 | 0.9778±0 | |

**Table C.83:** Summary of stability results on NIPS datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

| Problem | Weighted | Normal | p-value |
|---|---|---|---|
| arcene | 0.302±0.0602 | 0.258±0.055 | 0.207 |
| dexter | 0.1847±0.0259 | 0.2133±0.0422 | 0.3214 |
| dorothea | 0.0824±0.002 | 0.0828±0.0026 | 0.8466 |
| gisette | 0.048±0.0015 | 0.0483±0.0016 | 0.7459 |
| madelon | 0.2948±0.0083 | 0.2892±0.0124 | 0.3576 |

**Table C.84:** Summary of classification error results on NIPS datasets for **RLIW + 1R** with *sample*$\Delta$ combination. Significantly better results shown in bold face.

Appendix

# D List of publications


[1] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. *Double Relief with progressive weighting function*. Tech. rep. June 15, 2006. arXiv: 1509.04265. URL: http://arxiv.org/abs/1509.04265.

[2] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Exploiting the Accumulated Evidence for Gene Selection in Microarray Gene Expression Data." In: *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*. Ed. by Helder Coelho, Rudi Studer, and Michael Wooldridge. Vol. 215. Frontiers in Artificial Intelligence and Applications. IOS Press, 2010, pp. 989–990. DOI: 10.3233/978-1-60750-606-5-989. URL: http://dx.doi.org/10.3233/978-1-60750-606-5-989.

[3] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. *Exploiting the Accumulated Evidence for Gene Selection in Microarray Gene Expression Data*. Tech. rep. LSI-13-4-R. Universitat Politècnica de Catalunya, 2013. URL: http://www.cs.upc.edu/dept/techreps/llistat_detallat.php?id=1134.

[4] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Improved Stability of Feature Selection by Combining Instance and Feature Weighting." In: *Research and Development in Intelligent Systems XXXI*. Ed. by Max Bramer and Miltos Petridis. Springer International Publishing, 2014, pp. 35–49. ISBN: 978-3-319-12068-3. DOI: 10.1007/978-3-319-12069-0_3. URL: http://dx.doi.org/10.1007/978-3-319-12069-0_3.

[5] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Instance and Feature Weighted k-Nearest-Neighbors Algorithm." In: *ESANN 2016*. Submitted.

[6] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Remainder Subset Awareness for Feature Subset Selection." In: *Research and Development in Intelligent Systems XXVI, Incorporating Applications and Innovations in Intelligent Systems XVII, Peterhouse College, Cambridge, UK, 15-17 December 2009*. Ed. by Max Bramer, Richard Ellis, and Miltos Petridis. Springer, 2009, pp. 317–322. DOI: 10.1007/978-1-84882-983-1_25. URL: http://dx.doi.org/10.1007/978-1-84882-983-1_25.

[7] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Remainder subset awareness for feature subset selection." In: *IV Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA 2007)*. Ed. by P. Ibáñez et al. Red Española de Minería de Datos. Zaragoza, Spain, 2007, pp. 39–48.

[8] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. *Toward better feature weighting algorithms: a focus on Relief*. Tech. rep. June 15, 2005. arXiv: 1509.03755. URL: http://arxiv.org/abs/1509.03755.

# Bibliography


[1] U. Alon et al. "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays." In: *Proceedings of the National Academy of Sciences of the United States of America* 96.12 (June 8, 1999), pp. 6745–6750. ISSN: 0027-8424. DOI: 10.1073/pnas.96.12.6745. URL: http://dx.doi.org/10.1073/pnas.96.12.6745 (cit. on p. 48).

[2] A. Asuncion and D. J. Newman. *UCI Machine Learning Repository*. 2007. URL: http://mlearn.ics.uci.edu/MLRepository.html (cit. on pp. 45, 110).

[3] Ran G. Bachrach, Amir Navot, and Naftali Tishby. "Margin based feature selection - theory and algorithms." In: *Proceedings of the twenty-first international conference on Machine learning*. ICML '04. New York, NY, USA: ACM, 2004. ISBN: 1-58113-838-5. DOI: 10.1145/1015330.1015352. URL: http://dx.doi.org/10.1145/1015330.1015352 (cit. on pp. 38, 40, 108).

[4] Moshe Ben-Bassat. "Use of Distance Measures, Information Measures and Error Bounds in Feature Evaluation." In: *Handbook of Statistics*. Ed. by P. Krishnaiah and L. Kanal. Vol. 2. North Holland, 1982, pp. 773–791 (cit. on p. 13).

[5] V. Bolón-Canedo et al. "A review of microarray datasets and applied feature selection methods." In: *Information Sciences* 282 (Oct. 2014), pp. 111–135. ISSN: 00200255. DOI: 10.1016/j.ins.2014.05.042. URL: http://dx.doi.org/10.1016/j.ins.2014.05.042 (cit. on p. 105).

[6] Leo Breiman. "Bagging Predictors." In: *Machine Learning* 24.2 (1996), pp. 123–140. DOI: 10.1023/A:1018054314350 (cit. on pp. 34, 118).

[7] Leo Breiman. "Random Forests." In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/a%3a1010933404324 (cit. on p. 118).

[8] Leo Breiman et al. *Classification and Regression Trees*. Chapman & Hall/CRC, Jan. 1984. ISBN: 0412048418 (cit. on pp. 14, 16).

[9] Hua-Long Bu, Guo-Zheng Li, and Xue-Qiang Zeng. "Reducing error of tumor classification by using dimension reduction with feature selection." In: *The First International Symposium on Optimization and Systems Biology (OSB'07)*. Ed. by Xiang-Sun Zhang et al. Vol. 7. Lecture Notes in Operations Research. Beijing, China: World Publishing Corporation, Aug. 8-10, 2007, pp. 232–241. ISBN: 978-7-5062-7292-6/O568 (cit. on p. 105).

[10] Christopher J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition." In: *Data Min. Knowl. Discov.* 2.2 (1998), pp. 121–167 (cit. on p. 14).

[11] Herman Chernoff. "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations." In: *The Annals of Mathematical Statistics* 23.4 (Dec. 1952), pp. 493–507. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729330. URL: http://dx.doi.org/10.1214/aoms/1177729330 (cit. on p. 127).

[12] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks." In: *Machine Learning*. Vol. 20. 1995, pp. 273–297. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.9362 (cit. on pp. 37, 83).

[13] Koby Crammer et al. "Margin Analysis of the LVQ Algorithm." In: *In: Advances in Neural Information Processing Systems 2002*. 2002, pp. 462–469. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.3285 (cit. on pp. 37, 38).

[14] Robert H. Creecy et al. "Trading MIPS and memory for knowledge engineering." In: *Commun. ACM* 35.8 (1992), pp. 48–64. DOI: http://doi.acm.org/10.1145/135226.135228 (cit. on p. 15).

[15] Thomas G. Dietterich. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms." In: *Neural Computation* 10 (1998), pp. 1895–1923. DOI: 10.1162/089976698300017197 (cit. on p. 49).

[16] Peter Drotar and Zdenek Smekal. "Comparison of stability measures for feature selection." In: *2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Herl'any, Slovakia: IEEE, Jan. 2015, pp. 71–75. ISBN: 978-1-4799-8221-9. DOI: 10.1109/sami.2015.7061849. URL: http://dx.doi.org/10.1109/sami.2015.7061849 (cit. on pp. 22, 27).

[17] Sandrine Dudoit, Jane Fridlyand, and Terence P. Speed. "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data." In: *Journal of the American Statistical Association* 97.457 (2002), pp. 77–87. DOI: 10.1198/016214502753479248. eprint: http://pubs.amstat.org/doi/pdf/10.1198/016214502753479248 (cit. on p. 48).

[18] Kevin Dunne, Padraig Cunningham, and Francisco Azuaje. *Solutions to Instability Problems with Sequential Wrapper-Based Approaches To Feature Selection*. Tech. rep. TCD-CD-2002-28. Submitted to The Journal of Machine Learning Research, 2002. Dublin, Ireland: Department of Computer Science, Trinity College, 2002 (cit. on pp. 23, 33, 37).

[19] Bradley Efron. "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation." In: *Journal of the American Statistical Association* 78.382 (1983), pp. 316–331. DOI: 10.2307/2288636 (cit. on p. 34).

[20] Yoav Freund and Robert E. Schapire. "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting." In: (1997), pp. 119–139. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.8918 (cit. on p. 37).

[21] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. 2nd ed. Computer Science and Scientific Computing. San Diego, CA, USA: Academic Press Professional, Inc., Oct. 12, 1990. ISBN: 0-12-269851-7. URL: http://portal.acm.org/citation.cfm?id=92131 (cit. on pp. 56, 83).

[22] T. R. Golub et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring." In: *Science (New York, N.Y.)* 286.5439 (Oct. 15, 1999), pp. 531–537. ISSN: 0036-8075. DOI: 10.1126/science.286.5439.531. URL: http://dx.doi.org/10.1126/science.286.5439.531 (cit. on pp. 48, 105).

[23] Gavin J. Gordon et al. "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma." In: *Cancer research* 62.17 (Sept. 1, 2002), pp. 4963–4967. ISSN: 0008-5472. URL: http://view.ncbi.nlm.nih.gov/pubmed/12208747 (cit. on p. 48).

[24] Isabelle Guyon. *Design of experiments for the NIPS 2003 variable selection benchmark*. Tech. rep. July 2003. URL: http://clopinet.com/isabelle/Projects/NIPS2003/Slides/NIPS2003-Datasets.pdf (cit. on p. 47).

[25] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection." In: *J. Mach. Learn. Res.* 3 (2003), pp. 1157–1182. ISSN: 1533-7928 (cit. on pp. 42, 80).

[26] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines." In: 46.1-3 (2002), pp. 389–422. DOI: 10.1023/a%253a1012487302797. URL: http://dx.doi.org/10.1023/a%3a1012487302797 (cit. on p. 38).

[27] Isabelle Guyon et al. "Result Analysis of the NIPS 2003 Feature Selection Challenge." In: *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*. 2004, pp. 545–552. URL: http://papers.nips.cc/paper/2728-result-analysis-of-the-nips-2003-feature-selection-challenge (cit. on pp. 47, 110).

[28] Mark A. Hall. "Correlation-based Feature Selection for Machine Learning." PhD thesis. University of Waikato, 1999 (cit. on p. 13).

[29] Yue Han and Lei Yu. "A Variance Reduction Framework for Stable Feature Selection." In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, Dec. 2010, pp. 206–215. ISBN: 978-1-4244-9131-5. DOI: 10.1109/ICDM.2010.144. URL: http://dx.doi.org/10.1109/ICDM.2010.144 (cit. on pp. 7, 38, 109, 111).

[30] Rattikorn Hewett and Phongphun Kijsanayothin. "Tumor classification ranking from microarray data." In: *BMC Genomics* 9.2 (2008), pp. 1–11. ISSN: 1471-2164. DOI: 10.1186/1471-2164-9-s2-s21. URL: http://dx.doi.org/10.1186/1471-2164-9-s2-s21 (cit. on p. 105).

[31] Robert C. Holte. "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets." In: *Machine Learning* 11.1 (Apr. 1, 1993), pp. 63–90. ISSN: 0885-6125. DOI: 10.1023/a:1022631118932. URL: http://dx.doi.org/10.1023/a:1022631118932 (cit. on p. 118).

[32] Jin-Hyuk Hong and Sung-Bae Cho. "Cancer classification with incremental gene selection based on DNA microarray data." In: *Computational Intelligence in Bioinformatics and Computational Biology, 2008. CIBCB '08. IEEE Symposium on*. IEEE, 2008, pp. 70–74. ISBN: 978-1-4244-1778-0. DOI: 10.1109/cibcb.2008.4675761. URL: http://dx.doi.org/10.1109/cibcb.2008.4675761 (cit. on p. 105).

[33] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in Induction*. New York: Academic Press, 1966 (cit. on p. 17).

[34] George H. John, Ron Kohavi, and Karl Pfleger. "Irrelevant Features and the Subset Selection Problem." In: *Machine Learning, Proceedings of the Eleventh International Conference*. Ed. by William W. Cohen and Haym Hirsh. Rutgers University, New Brunswick, NJ, USA: Morgan Kaufmann, July 1994, pp. 121–129 (cit. on pp. 5, 11, 13, 81).

[35] Thomas Kailath. "The Divergence and Bhattacharyya Distance Measures in Signal Selection." In: *IEEE Transactions on Communications* 15.1 (Feb. 1967), pp. 52–60. ISSN: 0096-2244. DOI: 10.1109/tcom.1967.1089532. URL: http://dx.doi.org/10.1109/tcom.1967.1089532 (cit. on p. 128).

[36] Alexandros Kalousis, Julien Prados, and Melanie Hilario. "Stability of feature selection algorithms: a study on high-dimensional spaces." In: *Knowledge and Information Systems* 12.1 (Dec. 2006), pp. 95–116. ISSN: 0219-3116. DOI: 10.1007/s10115-006-0040-8 (cit. on pp. 23, 24, 26, 34, 35, 37).

[37] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, Jan. 2003. ISBN: 978-0-89871-546-0. DOI: 10.1137/1.9780898718898. URL: http://dx.doi.org/10.1137/1.9780898718898 (cit. on p. 130).

[38] Kenji Kira and Larry A. Rendell. "The Feature Selection Problem: Traditional Methods and a New Algorithm." In: *AAAI*. Cambridge, MA, USA: AAAI Press and MIT Press, 1992, pp. 129–134 (cit. on pp. 13, 19, 107).

[39] Ron Kohavi. "Feature Subset Selection as Search with Probabilistic Estimates." In: *AAAI Fall Symposium on Relevance*. Nov. 1994, pp. 122–126 (cit. on pp. 13, 93, 99).

[40] Ron Kohavi. *Wrappers for Performance Enhancements and Oblivious Decision Graphs*. Tech. rep. Stanford, CA, USA, 1995. URL: http://portal.acm.org/citation.cfm?id=892585 (cit. on p. 51).

[41] Ron Kohavi and George H. John. "Wrappers for feature subset selection." In: *Artificial Intelligence* 97.1-2 (Dec. 1997), pp. 273–324. ISSN: 0004-3702. DOI: 10.1016/s0004-3702(97)00043-x. URL: http://dx.doi.org/10.1016/s0004-3702(97)00043-x (cit. on pp. 13, 14, 20, 107).

[42] Daphne Koller and Mehran Sahami. "Toward Optimal Feature Selection." In: *ICML*. Ed. by Lorenza Saitta. San Francisco, CA: Morgan Kaufmann, 1996, pp. 284–292 (cit. on p. 55).

[43] Igor Kononenko. "Estimating Attributes: Analysis and Extensions of RELIEF." In: *ECML*. Ed. by Francesco Bergadano and Luc De Raedt. Vol. 784. Lecture Notes in Computer Science. Springer, 1994, pp. 171–182 (cit. on pp. 13, 20, 21, 68).

[44] Pavel Křížek, Josef Kittler, and Václav Hlaváč. "Improving Stability of Feature Selection Methods." In: *CAIP*. Ed. by Walter G. Kropatsch, Martin Kampel, and Allan Hanbury. Vol. 4673. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. Chap. 115, pp. 929–936. ISBN: 978-3-540-74271-5. DOI: 10.1007/978-3-540-74272-2_115 (cit. on pp. 23, 25, 26, 33, 36, 37).

[45] Ludmila I. Kuncheva. "A stability index for feature selection." In: *IASTED International Conference on Artificial Intelligence and Applications, part of the 25th Multi-Conference on Applied Informatics, Innsbruck, Austria, February 12-14, 2007*. Innsbruck, Austria: ACTA Press, 2007, pp. 390–395 (cit. on pp. 23–28, 33, 36, 110, 126).

[46] P. Langley. "Selection of relevant features in machine learning." In: *Proceedings of the AAAI Fall Symposium on Relevance*. New Orleans, LA, USA: AAAI Press, 1994, pp. 140–144 (cit. on p. 12).

[47] Huan Liu and Hiroshi Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Norwell, MA, USA: Kluwer Academic Publishers, 1998. ISBN: 0792381963. URL: http://portal.acm.org/citation.cfm?id=551943 (cit. on p. 99).

[48] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Available from http://www.inference.phy.cam.ac.uk/mackay/itila/. Cambridge University Press, 2003. URL: http://www.cambridge.org/0521642981 (cit. on p. 17).

[49] O. L. Mangasarian and W. H. Wolberg. "Cancer diagnosis via linear programming." In: 23.5 (1990), pp. 1–18 (cit. on p. 47).

[50] Aleix Martínez and Robert Benavente. *The AR Face Database*. Tech. rep. 24. Cites in Scholar Google: http://scholar.google.com/scholar?hl=en&lr=&client=firefox-a&cites=15042646876214698l2. Bellatera: Computer Vision Center, 1998. URL: http://www.cat.uab.cat/Public/Publications/1998/MaB1998 (cit. on p. 42).

[51] Brad L. Miller and David E. Goldberg. "Genetic Algorithms, Tournament Selection, and the Effects of Noise." In: *Complex Systems*. Vol. 9. 3. 1995, pp. 193–212. URL: http://www.complex-systems.com/pdf/09-3-2.pdf (cit. on p. 130).

[52] T. Mohri and H. Tanaka. *An optimal weighting criterion of case indexing for both numeric and symbolic attributes*. 1994 (cit. on p. 16).

[53] R. Paredes and E. Vidal. "Learning weighted metrics to minimize nearest-neighbor classification error." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.7 (July 5, 2006), pp. 1100–1110. ISSN: 0162-8828. DOI: 10.1109/tpami.2006.145. URL: http://dx.doi.org/10.1109/tpami.2006.145 (cit. on p. 7).

[54] Roberto Paredes and Enrique Vidal. "Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization." In: *Pattern Recognition* 39.2 (Feb. 2006), pp. 180–188. ISSN: 00313203. DOI: 10.1016/j.patcog.2005.06.001. URL: http://dx.doi.org/10.1016/j.patcog.2005.06.001 (cit. on p. 7).

[55] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Series In Representation And Reasoning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988 (cit. on p. 54).

[56] Gabriel Prat Masramon and Lluís A. Belanche Muñoz. "Improved Stability of Feature Selection by Combining Instance and Feature Weighting." In: *Research and Development in Intelligent Systems XXXI*. Ed. by Max Bramer and Miltos Petridis. Springer International Publishing, 2014, pp. 35–49. ISBN: 978-3-319-12068-3. DOI: 10.1007/978-3-319-12069-0_3. URL: http://dx.doi.org/10.1007/978-3-319-12069-0_3 (cit. on p. 115).

[57] Pavel Pudil, Jana Novovicová, and Josef Kittler. "Floating search methods in feature selection." In: *Pattern Recognition Letters* 15.11 (1994), pp. 1119–1125 (cit. on pp. 14, 33).

[58] P. Pudil et al. "Floating search methods for feature selection with nonmonotonic criterion functions." In: *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on Pattern Recognition*. Vol. 2. IAPR. Nov. 1994, pp. 279–283 (cit. on p. 13).

[59] Ross J. Quinlan. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993 (cit. on p. 17).

[60] Ross J. Quinlan. "Induction of Decision Trees." In: *Machine Learning* 1.1 (1986), pp. 81–106 (cit. on p. 17).

[61] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2008. ISBN: 3-900051-07-0 (cit. on pp. 83, 98).

[62] S. Ramaswamy et al. "Multiclass cancer diagnosis using tumor gene expression signatures." In: *Proceedings of the National Academy of Sciences of the United States of America* 98.26 (Dec. 18, 2001), pp. 15149–15154. ISSN: 0027-8424. DOI: 10.1073/pnas.211566398. URL: http://dx.doi.org/10.1073/pnas.211566398 (cit. on pp. 48, 105).

[63] Šarunas Raudys, Richard Baumgartner, and Ray Somorjai. "On Understanding and Assessing Feature Selection Bias." In: *Artificial Intelligence in Medicine*. Ed. by David Hutchison et al. Vol. 3581. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. Chap. 63, pp. 468–472. ISBN: 978-3-540-27831-3. DOI: 10.1007/11527770_63. URL: http://dx.doi.org/10.1007/11527770_63 (cit. on p. 49).

[64] Roberto Ruiz, José C. Riquelme, and Jesús S. Aguilar-Ruiz. "Incremental wrapper-based gene selection from microarray data for cancer classification." In: *Pattern Recognition* 39.12 (Dec. 2006), pp. 2383–2392. ISSN: 00313203. DOI: 10.1016/j.patcog.2005.11.001. URL: http://dx.doi.org/10.1016/j.patcog.2005.11.001 (cit. on p. 99).

[65] Yvan Saeys, Thomas Abeel, and Yves Peer. "Robust Feature Selection Using Ensemble Feature Selection Techniques." In: *ECML PKDD '08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*. Antwerp, Belgium: Springer-Verlag, 2008, pp. 313–325. ISBN: 978-3-540-87480-5. DOI: 10.1007/978-3-540-87481-2_21 (cit. on pp. 34, 36, 37).

[66] Robert E. Schapire. "A Brief Introduction to Boosting." In: *16th International Joint Conference on Artificial Intelligence*. Ed. by Thomas Dean. Vol. 2. IJCAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., July 31, 1999, pp. 1401–1406. ISBN: 1-55860-613-0 (cit. on p. 14).

[67] Federico Schlüter. "A survey on independence-based Markov networks learning." In: *Artificial Intelligence Review* 42.4 (June 21, 2014), pp. 1069–1093. ISSN: 0269-2821. DOI: 10.1007/s10462-012-9346-y. URL: http://dx.doi.org/10.1007/s10462-012-9346-y (cit. on p. 131).

[68] C. E. Shannon. "A Mathematical Theory of Communication." In: *Bell System Technical Journal* 27.4 (Oct. 1948), pp. 623–656. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb00917.x. URL: http://dx.doi.org/10.1002/j.1538-7305.1948.tb00917.x (cit. on pp. 16, 25).

[69] Claude E. Shannon, Warren Weaver, and Norbert Wiener. *The Mathematical Theory of Communication*. Vol. 3. 9. Urbana: University of Illinois Press, 1950, pp. 31+. DOI: 10.1063/1.3067010. URL: http://dx.doi.org/10.1063/1.3067010 (cit. on p. 16).

[70] Marko R. Šikonja and Igor Kononenko. "Theoretical and Empirical Analysis of ReliefF and RReliefF." In: *Machine Learning* 53.1-2 (2003), pp. 23–69. ISSN: 0885-6125. DOI: 10.1023/A:1025667309714. URL: http://dx.doi.org/10.1023/A:1025667309714 (cit. on pp. 21, 66).

[71] Dinesh Singh et al. "Gene expression correlates of clinical prostate cancer behavior." In: *Cancer cell* 1.2 (Mar. 2002), pp. 203–209. ISSN: 1535-6108. URL: http://view.ncbi.nlm.nih.gov/pubmed/12086878 (cit. on p. 48).

[72] Surendra K. Singhi and Huan Liu. "Feature subset selection bias for classification learning." In: *the 23rd international conference*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM Press, 2006, pp. 849–856. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143951. URL: http://dx.doi.org/10.1145/1143844.1143951 (cit. on p. 49).

[73] Petr Somol and Jana Novovičová. "Evaluating the Stability of Feature Selectors That Optimize Feature Subset Cardinality." In: *Lecture Notes in Computer Science*. Ed. by Niels Vitoria Lobo et al. Vol. 5342. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Chap. 99, pp. 956–966. ISBN: 978-3-540-89688-3. DOI: 10.1007/978-3-540-89689-0_99 (cit. on pp. 23, 26, 36).

[74] Craig Stanfill and David Waltz. "Toward memory-based reasoning." In: *Communications of the ACM* 29.12 (Dec. 1, 1986), pp. 1213–1228. ISSN: 0001-0782. DOI: 10.1145/7902.7906. URL: http://dx.doi.org/10.1145/7902.7906 (cit. on pp. 16, 61).

[75] S. D. Stearns. "On selecting features for pattern classifiers." In: *Proceedings of the 3rd International Conference on Pattern Recognition (ICPR 1976)*. Coronado, CA, 1976, pp. 71–75 (cit. on p. 14).

[76] Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, eds. *Advances in Neural Information Processing Systems 16. Proceedings of the 2003 Conference*. illustrated edition. London, England: The MIT Press, June 4, 2004. ISBN: 0-262-20152-6. URL: http://www.worldcat.org/isbn/0262201526 (cit. on pp. 47, 110).

[77] Andrey N. Tikhonov and Vasiliy I. Arsenin. *Solutions of ill-posed problems*. Ed. by Fritz John. Scripta series in mathematics. New York: V.H. Winston & Sons, 1977. ISBN: 9780470991244 (cit. on p. 4).

[78] C. J. Van Rijsbergen. *Information Retrieval*. 2nd. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN: 978-0408709293. URL: http://www.worldcat.org/isbn/0408709294 (cit. on p. 37).

[79] Laura J. van 't Veer et al. "Gene expression profiling predicts clinical outcome of breast cancer." In: *Nature* 415.6871 (Jan. 31, 2002), pp. 530–536. ISSN: 0028-0836. DOI: 10.1038/415530a. URL: http://dx.doi.org/10.1038/415530a (cit. on p. 48).

[80] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Ed. by J. Chambers et al. Fourth Edition. Statistics and Computing. New York, NY: Springer New York, 2002. ISBN: 978-1-4419-3008-8. DOI: 10.1007/978-0-387-21706-2. URL: http://www.stats.ox.ac.uk/pub/MASS4 (cit. on pp. 83, 98).

[81] L. Wang, J. Zhu, and H. Zou. "Hybrid huberized support vector machines for microarray classification and gene selection." In: *Bioinformatics* 24.3 (Jan. 5, 2008), pp. 412–419. ISSN: 1460-2059. DOI: 10.1093/bioinformatics/btm579. URL: http://dx.doi.org/10.1093/bioinformatics/btm579 (cit. on p. 105).

[82] Dietrich Wettschereck, David W. Aha, and Takao Mohri. "A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms." In: *Artif. Intell. Rev.* 11.1-5 (1997), pp. 273–314 (cit. on p. 67).

[83] D. Randall Wilson and Tony R. Martinez. "Improved Heterogeneous Distance Functions." In: *J. Artif. Int. Res.* 6.1 (Jan. 1997), pp. 1–34. ISSN: 1076-9757. DOI: 10.1613/jair.346. URL: http://dx.doi.org/10.1613/jair.346 (cit. on pp. 19, 61).

[84] David Wolpert and William G. Macready. "No free lunch theorems for optimization." In: *IEEE Transactions on Evolutionary Computation* 1.1 (Apr. 1997), pp. 67–82. ISSN: 1089-778X. DOI: 10.1109/4235.585893. URL: http://dx.doi.org/10.1109/4235.585893 (cit. on p. 91).

[85] L. Xu et al. "Optimized sample-weighted partial least squares." In: *Talanta* 71.2 (Feb. 15, 2007), pp. 561–566. ISSN: 00399140. DOI: 10.1016/j.talanta.2006.04.039. URL: http://dx.doi.org/10.1016/j.talanta.2006.04.039 (cit. on p. 129).

[86] Xulei Yang, Qing Song, and Aize Cao. "Weighted support vector machine for data classification." In: *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*. Vol. 2. IEEE, July 2005, 859–864 vol. 2. ISBN: 0-7803-9048-2. DOI: 10.1109/ijcnn.2005.1555965. URL: http://dx.doi.org/10.1109/ijcnn.2005.1555965 (cit. on p. 129).