Visualization and Interpretability in Probabilistic Dimensionality Reduction Models

Alessandra Tosi

Department of Software

Universitat Politècnica de Catalunya

This dissertation is submitted for the degree of

Doctor of Philosophy

2014

Abstract

Over the last few decades, data analysis has swiftly evolved from being a task addressed mainly within the remit of multivariate statistics, to an *endevour* in which data heterogeneity, complexity and even sheer size, driven by computational advances, call for alternative strategies, such as those provided by pattern recognition and machine learning.

Any data analysis process aims to extract new knowledge from data. Knowledge extraction is not a trivial task and it is not limited to the generation of data models or the recognition of patterns. The use of machine learning techniques for multivariate data analysis should in fact aim to achieve a dual target: interpretability and good performance. At best, both aspects of this target should not conflict with each other. This gap between data modelling and knowledge extraction must be acknowledged, in the sense that we can only extract knowledge from models through a process of interpretation.

Exploratory information visualization is becoming a very promising tool for interpretation. When exploring multivariate data through visualization, high data dimensionality can be a big constraint, and the use of dimensionality reduction techniques is often compulsory. The need to find flexible methods for data modelling has led to the development of non-linear dimensionality reduction techniques, and many state-of-the-art approaches of this type fall in the domain of probabilistic modelling. These non-linear techniques can provide a flexible data representation and a more faithful model of the observed data compared to the linear ones, but often at the expense of model interpretability, which has an impact in the model visualization results.

In manifold learning non-linear dimensionality reduction methods, when a high-dimensional space is mapped onto a lower-dimensional one, the obtained embedded manifold is subject to local geometrical distortion induced by the non-linear mapping. This kind of distortion can often lead to misinterpretations of the data set structure and of the obtained patterns. It is important to give relevance to the problem of how to quantify and visualize the distortion itself in order to interpret

data in a more faithful way.

The research reported in this thesis focuses on the development of methods and techniques for explicitly reintroducing the local distortion created by non-linear dimensionality reduction models into the low-dimensional visualization of the data that they produce, as well as in the definition of metrics for probabilistic geometries to address this problem. We do not only provide methods only for static data, but also for multivariate time series.

The reintegration of the quantified non-linear distortion into the visualization space of the analysed non-linear dimensionality reduction methods is a goal by itself, but we go beyond it and consider alternative adequate metrics for probabilistic manifold learning. For that, we study the role of *Random geometries*, that is, distributions of manifolds, in machine learning and data analysis in general. Methods for the estimation of distributions of data-supporting Riemannian manifolds as well as algorithms for computing interpolants over distributions of manifolds are defined. Experimental results show that inference made according to the random Riemannian metric leads to a more faithful generation of unobserved data.

Abstract

(Catalan)

Durant les últimes dècades, l'anàlisi de dades ha evolucionat ràpidament de ser una tasca dirigida principalment dins de l'àmbit de l'estadística multivariant, a un *endevour* en el qual l'heterogeneïtat de les dades, la complexitat i la simple grandària, impulsats pels avanços computacionals, exigeixen estratègies alternatives, tals com les previstes en el Reconeixement de Formes i l'Aprenentatge Automàtic.

Qualsevol procés d'anàlisi de dades té com a objectiu extreure nou coneixement a partir de les dades. L'extracció de coneixement no és una tasca trivial i no es limita a la generació de models de dades o el reconeixement de patrons. L'ús de tècniques d'aprenentatge automàtic per a l'anàlisi de dades multivariades, de fet, hauria de tractar d'aconseguir un objectiu doble: la interpretabilitat i un bon rendiment. En el millor dels casos els dos aspectes d'aquest objectiu no han d'entrar en conflicte entre sí. S'ha de reconèixer la bretxa entre el modelatge de dades i l'extracció de coneixement, en el sentit que només podem extreure coneixement a partir dels models a través d'un procés d'interpretació.

L'exploració de la visualització d'informació s'està convertint en una eina molt prometedora per a la interpretació dels models. Quan s'exploren les dades multivariades a través de la visualització, la gran dimensionalitat de les dades pot ser un obstacle, i moltes vegades és obligatori l'ús de tècniques de reducció de dimensionalitat. La necessitat de trobar mètodes flexibles per al modelatge de dades ha portat al desenvolupament de tècniques de reducció de dimensionalitat no lineals. L'estat de l'art d'aquests enfocaments cau moltes vegades en el domini de la modelització probabilística. Aquestes tècniques no lineals poden proporcionar una representació de les dades flexible i un model de les dades més fidel comparades amb els models lineals, però moltes vegades a costa de la interpretabilitat del model, que té un impacte en els resultats de visualització.

En els mètodes d'aprenentatge de varietats amb reducció de dimensionalitat no lineals, quan un espai d'alta dimensió es projecta sobre un altre de dimensió menor, la varietat immersa obtinguda està subjecta a una distorsió geomètrica local induïda

per la funció no lineal. Aquest tipus de distorsió pot conduir a interpretacions errònies de l'estructura del conjunt de dades i dels patrons obtinguts. Per això, és important donar rellevància al problema de com quantificar i visualitzar aquesta distorsió en sí, amb la finalitat d'interpretar les dades d'una manera més fidel.

La recerca presentada en aquesta tesi se centra en el desenvolupament de mètodes i tècniques per reintroduir de forma explícita a l'espai de visualització la distorsió local creada per la funció no lineal. Aquesta recerca se centra també en la definició de mètriques per a geometries probabilístiques per fer front al problema de la distorsió de la funció en els models de reducció de dimensionalitat no lineals. No proporcionem mètodes només per a les dades estàtiques, sinó també per a sèries temporals multivariades.

La reintegració de la distorsió no lineal a l'espai de visualització dels mètodes de reducció de dimensionalitat no lineals analitzats és un objectiu en sí mateix, però aquesta anàlisi va més enllà i considera també les mètriques probabilístiques adequades a l'aprenentatge de varietats probabilístiques. Per això, estudiem el paper de les Geometries Aleatòries (distribucions de les varietats) en Aprenentatge Automàtic i anàlisi de dades en general. Es defineixen aquí els mètodes per a l'estimació de les distribucions de varietats de Riemann de suport a les dades, així com els algorismes per calcular interpolants en les distribucions de varietats. Els resultats experimentals mostren que la inferència feta segons les mètriques de les varietats Riemannianes Aleatòries dóna origen a una generació de les dades observades més fidel.

Alla mia famiglia.

Acknowledgements

First I would like to acknowledge my advisor Alfredo Vellido for the great dedication that he put in the supervision of this thesis: thanks to his patient and guidance I have been able to reach this achievement. When I started this journey I had no experience as a researcher, but since the beginning he has shown interest in my opinion, making me feel as an active part of the research group. Alfredo has always been open to new ideas coming from his students and he encouraged me to think out of the schemes. He has been present as a supervisor, adapting to my rhythm, even when I needed some last-minute revision under deadlines. For these and other reasons I will tell him a big Thank You.

I would also thank Neil Lawrence for his supervision and great help during my research stay in Sheffield. His priceless guidance had a great impact in my student path, stimulating me to reach challenging goals. From my first day in his group Neil made me feel like home, and his passion and enthusiasm (not only in research, but in all aspects of life) made my time in Sheffield to be an unforgettable life experience. Thank You Neil.

Thanks to my co-authors for their collaboration and help. Thank to Lluís Belanche for giving value to my ideas when I was just a first year (actually, first-days) PhD student. Thanks to Ivan Olier for the patience and attention that he put in our collaboration. A special Thank You to Søren Hauberg, his enlightening ideas and his exceptional enthusiasm have been fundamental to develop our joint work; I am looking forward to sit around a table and discuss with you about extraordinary mathematical theories.

I acknowledge the projects that partially founded my research: FP7 HEALTH 2013.2.4.2-1 Shock-omics, TIN2012-31377 - Kappaaim, and TIN2009 13895-C02-01 - AIDTumour.

Thank you to the colleagues that started with me this journey and made my experience a lot more fun. Thanks to my group-mates Sol and Albert. Thanks to all the crazy and great people who populates the floor S1 of the CS department in Barcelona, thanks for the many lunches and coffees together. Thanks to Sign'n'Forget for the acrobatic problem solving. Thanks to all the people that passed from SITraN during the last months, you made me not want to leave Sheffield (and, in fact, I did not). Grazie ad Alessandro per la pazienza (infinita) e per essere stato un amico prezioso e grazie a Luca per i tanti caffè al Blati e le lunghe chiacchierate. Grazie anche alla Maestra Anna, che mi ha fatto amare la matematica sin dal principio del mio percorso educativo, ed al Professor Bigoni, che ha accresciuto la mia passione per la scienza.

The last thanks goes to Andreas, for his patience and support. Thank you for reading the whole thesis, thank you for standing my crazy moments, but mostly thanks for being by my side donating me a smile every day.

Dedico questa tesi a coloro che hanno contribuito maggiormente al raggiungimento di questo obiettivo: la mia famiglia. Senza di voi non sarei mai arrivata a questo punto, mi avete sostenuta in ogni modo possibile e mi avete costantemente incoraggiata e spronata, nonché aiutata e consolata nei momenti difficili. Tutto ciò che ho ottenuto finora e che vedo per il mio futuro lo devo a voi, GRAZIE! Grazie a mamma Giulia, papà Carlo e mio fratello Raffaele, che è e sempre sarà un insostituibile punto di riferimento.

Contents

List of Figures

Chapter 1

Introduction

Over the last few decades, data analysis has swiftly evolved from being a task addressed mainly within the remit of multivariate statistics, to an *endevour* in which data heterogeneity, complexity and even sheer size, driven by computational advances, call for alternative strategies, such as those provided by pattern recognition and machine learning.

These new data requirements, nowadays under the fashionable concept of *Big data*, come not only from business enterprises as an extension of traditional data mining, but also from scientific fields such as, for instance, biology [Marx, 2013]. The ensuing big challenge for pattern recognition and machine learning is the translation of raw data into useful knowledge that can be acted upon in practical terms.

Any data analysis process, in the end, aims to extract new knowledge from data. Knowledge extraction is not a trivial task and it is not limited to the generation of data models (regardless their sophistication) or to the recognition of (possibly relevant) patterns. Those patterns and models and, in fact, any other results stemming from our analyses require interpretation to become knowledge.

Consequently, the use of machine learning techniques for multivariate data (MVD) analysis should aim to achieve a dual target: interpretability and good performance. At best, both aspects of this target should not conflict with each other. A gap between data modeling and knowledge extraction must thus be acknowledged, in the sense that we can only extract knowledge from models through a process of interpretation [Vellido et al., 2012]. Models are often built with the sole goal of achieving high accuracy or precision, even though in many practical applications an optimum performance is likely to be less relevant than achieving interpretability.

In this context, exploratory information visualization becomes an useful tool. When exploring MVD through visualization, high data dimensionality can be a big

constraint, and the use of dimensionality reduction techniques becomes almost compulsory.

Dimensionality reduction techniques are in fact a key tool in high-dimensional MVD analysis, and a large corpus of literature addressing this problem (mostly from the viewpoints of feature selection and feature extraction) is currently available to us, tracing back to over a century ago. The best known and most widely used linear feature extraction dimensionality reduction method is Principal Component Analysis (PCA), introduced by Pearson [1901]. In essence, PCA assumes that a low dimensional latent space, with Gaussian distributed variables, is mapped into the observed data space under a linear transformation. The reduction of dimensionality is operated by finding a few orthogonal linear combinations (principal components) of the original variables with the largest variance. The key to the resilience of this method after more than a century is, probably, its easy interpretability, as the extracted features are just linear combinations of the original ones in the data set.

The need to find more flexible (and hopefully better performing) methods for MVD modeling has led to the development of non-linear techniques for dimensionality reduction, which are slowly growing in popularity [Lee and Verleysen, 2007]. A modern approach to non-linear dimensionality reduction (NLDR) of relevance to the current thesis and that involves probabilistic modeling (and, therefore, statistical machine learning) is latent variable modeling (LVM), which works by defining a subset of latent (or hidden) variables to accompany and explain the observed ones. NLDR methods can provide a flexible data representation and a more faithful model of the observed MVD than linear ones. This target is too often reached at the expense of model interpretability, which has an impact in the model visualization results.

Both linear and non-linear DR methods aim, in one way or another, to preserve the structure of the observed data as much as possible in the low dimensional data representation that they generate. Unfortunately (from the point of view of interpretation) NLDR methods usually generate different levels of mapping distortion, geometrical and topological, including: manifold compression, stretching, gluing and tearing [Aupetit, 2007]. Many distortion measures (often associated with specific models and specific visualization techniques) have been proposed for different NLDR methods.

In manifold learning, when a high-dimensional space is mapped onto a lower-dimensional one, the obtained embedded manifold is subject to some kind of local geometrical distortion induced by the non-linear mapping. This means that there is no guarantee that the inter-point distances in the observed data space will be uniformly reflected in the visualization space. This kind of distortion can often lead

to misinterpretations of the data set itself.

At best, these NLDR methods can aspire to minimize the distortion of the observed data introduced in their representation, according to some objective function. But, given that it is almost impossible to completely avoid geometrical distortions while reducing dimensionality, it is important to give relevance to another aspect of the problem: how to quantify and visualize this distortion itself in order to interpret data in a more faithful way.

The research reported in this thesis focuses on the development of methods and techniques for explicitly reintroducing the local distortion created by NLDR models into the low-dimensional representation of the MVD for visualization that they produce, as well as on the definition of metrics for probabilistic geometries to address this problem.

For part of this research, we draw inspiration from a technique originally devised for the analysis of geographic information, namely density-equalizing maps, or Cartograms [Gastner and Newman, 2004]. These maps were originally defined as geographic maps in which the sizes of delimited regions appear distorted in proportion to underlying quantities such as their population. Cartograms were later redefined, using diffusion techniques from physics, to avoid drawbacks such as the undesired overlapping of regions, or a too strong dependence on the choice of coordinate axes.

The interpretation leap in the use of Cartograms for NLDR model visualization consists on extrapolating from geographical maps to the latent visualization spaces of NLDR models (particularly manifold learning methods in this thesis), as well as on substituting geography-distorting quantities such as population density by quantities reflecting the mapping distortion introduced by the non-linear methods.

We do not aim to provide methods only for static data in the thesis, but also for multivariate time series (MTS). Again, MTS visualization may become difficult to interpret when data are modelled using non-linear techniques. This is the case, for instance, when modelled using Variational Bayesian Generative Topographic Mapping Through Time (VB-GTM-TT) [Olier and Vellido, 2008a], a variational Bayesian variant of the manifold learning family defined for MTS visualization. Its interpretability will be improved through the explicit estimation of probabilities of transition between states described in the visualization space and the quantification of the non-linear mapping distortion.

The reintegration of the quantified non-linear distortion into the visualization space of the analysed NLDR methods is a goal by itself, but we want to go beyond that and consider alternative adequate metrics for probabilistic manifold learning.

To accomplish this, we study the role of Random Geometries, that is, distributions of manifolds in machine learning and data analysis in general. Methods for the estimation of distributions of data-supporting Riemannian manifolds, as well as algorithms for computing interpolants (geodesics) over distributions of manifolds are defined.

In this thesis we propose methods to increase the interpretability of NLDR methods in different instances of MVD analysis using visualization. It is important to stress that this analysis could quite straightforwardly be extended not only to other variants of the methods we investigate (variants of Self-Organizing Maps (SOM) [Kohonen, 2001], Generative Topographic Mapping (GTM) [Bishop et al., 1998a; Svensén, 1998] and Gaussian Process LVM (GP-LVM) [Lawrence, 2005]), but also to other alternative NLDR visualization-oriented methods, provided a local distortion measure, or some approximation for it, could be calculated.

1.1 Summary of the main goals of the thesis

The generic goals of the current thesis could be summarily listed as follows:

- *GG1*: Exploration of the concept of local mapping distortion in non-linear dimensionality reduction methods (with specific attention paid to manifold learning techniques) from the viewpoint of the analytical quantification of such distortion.

- *GG2*: Exploration of the Cartogram representation in bounded and partitioned visualization spaces as a tool for increasing the interpretability and usability of such multivariate data visualizations. Definition and implementation of Cartogram-based algorithms for visual representation of multivariate data for batch-SOM and GTM, based on magnification factor (MF) measurements of the mapping distortion they generate.

- *GG3*: Explicit estimation of probabilities of transition between states described in the visualization space and quantification of the non-linear mapping distortion for VB-GTM-TT in the analysis of multivariate time series.

- *GG4*: Definition of adequate metrics for probabilistic manifold learning as an alternative to the standard Euclidean metrics through the study of Random Geometries, including definition of methods for the estimation of distributions of data-supporting Riemannian manifolds as well as algorithms for computing interpolants over distributions of manifolds.

1.2 Structure of the document

The thesis document is structured in the following chapters:

Chapter 1

The document starts with a general introduction to the field of interest, mentioning problems that will be tackled later on in this work. We then provide a summary of the notation and symbols used over the document.

Chapter 2

This chapter provides the introductory technical background about probabilistic data modelling with a focus on non-linear dimensionality reduction methods for multivariate data visualization. We devote some special attention to manifold learning techniques such as the Generative Topographic Mapping (GTM) and the Gaussian Process Latent Variable Model (GP-LVM).

Chapter 3

This is again a technical background chapter, which focuses on the general theme of distortion measures in non-linear dimensionality reduction methods, paying special attention to the concept of Magnification Factors. We also include some necessary basics about Riemannian geometry.

Chapter 4

In this chapter, we present research results in the topic of Cartogram-based visualization of multivariate data using manifold learning models. This includes self-contained introductions to Cartogram methods and to Self-Organizing Maps (SOM) models in different variants. We show how to apply Cartograms to the representations of non-linear dimensionality reduction distortion measures in the visualization space with experiments including batch-SOM and t-GTM.

Chapter 5

Our analysis departs from static i.i.d. data to address methods to improve visualization-based analysis of multivariate time series using dynamic variants of manifold learning models. This chapter includes a self-contained definition of Variational Bayesian GTM through time (VB-GTM-TT) and an experimental set.

Chapter 6

It provides a study of adequate metrics for probabilistic geometries and their

impact on model interpretability. It includes a definition of a probabilistic Riemannian metric for GP-LVM and algorithms for the calculation of geodesics distances for this model. A battery of experiments to evaluate the proposed methods is reported.

Chapter 7

The final chapter summarises some conclusions of the thesis, highlighting its novelties. It also includes and lists some advanced themes and expected potential future avenues of research that we envisage beyond the advances presented in the thesis.

1.3 Refereed publications directly related to the thesis

[1] A. Tosi, S. Hauberg, A. Vellido, N.D. Lawrence. *Metrics for probabilistic geometries.* In The 30^{th} Conference on Uncertainty in Artificial Intelligence (UAI 2014), pp. 800–808. Quebec City, Canada.

[2] A. Tosi, A. Vellido. *Probabilistic Geometries as a tool for Interpretability in Dimensionality Reduction Models.* The 8^{th} WiML Workshop, Advances in Neural Information Processing Systems (NIPS 2014). Montreal, Canada.

[3] A. Tosi and A. Vellido. *Local metric and graph based distance for probabilistic dimensionality reduction.* The Workshop on Features and Structures (FEAST 2014) International Conference on Pattern Recognition (ICPR 2014), Stockholm, Sweden.

[4] A. Tosi, I. Olier, A. Vellido. *Probability ridges and distortion flows: Visualizing multivariate time series using a variational Bayesian manifold learning method.* In Advances in Intelligent Systems and Computing, Vol.295, pp.55-64, procs. of the 10^{th} Workshop on Self-Organizing Maps (WSOM 2014), Mittweida, Germany.

[5] A. Tosi, A. Vellido. *Robust cartogram visualization of outliers in manifold learning.* In Proceedings of the 21^{st} European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), Bruges, Belgium, pp.555-560.

[6] A. Tosi, A. Vellido. *Cartogram representation of the batch-SOM magnification factor.* In Proceedings of the 20^{th} European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), Bruges, Belgium, pp.203-208.

Description of contributions in relation to the refereed publications

In **Chapter 3** we introduce the Cartogram-based method which has been presented in [6] and [5] for the batch-SOM algorithm and the t-GTM respectively; in Sec. 4.3 and Sec. 4.4 we illustrate the experimental results as reported in the two papers.

The analysis on multivariate time series reported in [4] is presented in **Chapter 5**, together with the experimental results; the design and implementation of the algorithms has been done using software tools for VB-GTM-TT [Olier and Vellido, 2008b] provided by Dr. Iván Olier.

The idea of probabilistic geometries presented in **Chapter 6** is the result of a joint project carried out at the Machine Learning group of the University of Sheffield, with Professor Neil Lawrence. The experimental results of [1] are reported in Sec. 6.4. The tools used to design the experiments and train the models are built on the GP-LVM [Lawrence, 2005] software provided by the Machine Learning group of the University of Sheffield[1]; the tools used to compute manifold structures and geodesics via ODE's solutions, as described in 6.3.2, use the software provided by Dr. Søren Hauberg and previously used in [Hauberg et al., 2012]. The description of the geodesic computation via discrete graphs presented in 6.3.1 refers to [3].

Finally, the work presented in [2] provides an overall summary of the topics of this thesis, focusing on the problem of interpretability in dimensionality reduction introduced in **Chapter 2** and **Chapter 3** and presenting the advances detailed in **Chapter 6**.

1.4 Other refereed publications

It is worth mentioning other publications that, although not included in the thesis, have inspired its development while exploring new and interesting topics of research:

[7] L. A. Belanche, A. Tosi. *Averaging of Kernel Functions. Neurocomputing* **112** (2013), pp.19-25.

[1]Software can be downloaded here: https://github.com/SheffieldML/, both in the Matlab and in the Python version

[8] L. A. Belanche, A. Tosi. *Averaging of kernel functions.* In Proceedings of the 20^{th} European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), Bruges, Belgium, pp.363-368.

1.5 Symbols and notation

- Notation:

The matrix $\mathbf{Y} \in \mathbb{R}^{N \times p}$ represents the observed data space, where each row corresponds to an observed data point and each column to a dimension. We denote with $\mathbf{Y}_{:,j}$ the columns of the data matrix, with $\mathbf{y}_i$ the rows of the data matrix and with $y_{i,j}$ a single scalar element.

$$\mathbf{Y} = [\mathbf{Y}_{:,1}\mathbf{Y}_{:,2}\dots\mathbf{Y}_{:,p}] = \overbrace{\left[\begin{array}{ccccc} y_{1,1} & y_{1,2} & \cdot & \cdot & y_{1,p} \\ y_{2,1} & y_{2,2} & \cdot & \cdot & y_{2,p} \\ \cdot & & & & \\ \cdot & & & & \\ y_{N,1} & y_{N,2} & \cdot & \cdot & y_{N,p} \end{array}\right]}^{\text{data features}} \Bigg\} \text{data points}$$

Similarly, we denote with $\mathbf{x}_i$ the rows of the matrix $\mathbf{X}$.

In this document we use the symbol $\mathbf{x}$ to represent a latent vector of dimension q, and the symbol $\mathbf{y}$ to represent an observed vector (data point) of dimension p. We always consider $q < p$.

Given a differentiable function

$$f : \mathbb{R}^q \longrightarrow \mathbb{R}^p$$
$$\mathbf{x} \mapsto \mathbf{y} = f(\mathbf{x})$$

we call Jacobian the $p \times q$ matrix $\mathbf{J}$ containing all the partial derivatives

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y^{(1)}}{\partial x^{(1)}} & \cdots & \frac{\partial y^{(1)}}{\partial x^{(q)}} \\ \vdots & & \vdots \\ \frac{\partial y^{(p)}}{\partial x^{(1)}} & \cdots & \frac{\partial y^{(p)}}{\partial x^{(q)}} \end{bmatrix} \tag{1.1}$$

- Symbols:

$\mathbb{R}$	the set of real numbers
$\mathbf{I}$	identity matrix
$\mathbf{\Phi}(\cdot)$	a vector of function values, the mth element corresponds to $\phi_m(\cdot)$
$\mathbf{f}(\mathbf{X})$	a vector of function values, the ith element corresponds to $f(\mathbf{x}_i)$.
$\mathbf{K}_{\mathbf{f},\mathbf{f}}$	covariance matrix whose elements are given by $k(\mathbf{x}_i, \mathbf{x}_j)$
$x^{(i)}$	the ith component of the vector $\mathbf{x}$
$\frac{\partial}{\partial x^{(i)}}$	the partial derivative with respect to $x^{(i)}$
$\mathbf{J}$	the Jacobian of a function
$\bigtriangledown^2$	the Laplacian operator of a function
$\sim$	distributed according to the following probability distribution
$\mathcal{GP}(\cdot, \cdot)$	Gaussian Process
$\mathcal{N}(\cdot, \cdot)$	Gaussian Distribution
$\Gamma(\cdot, \cdot)$	Gamma Distribution
$\mathbb{E}[x]$	expectation of the random variable x

- Acronyms:

BMU	Best Matching Unit
BSOM	Batch Self Organizing Maps
DM	Data Mining
EQ	Exponentiated Quadratic kernel
GP	Gaussian Process
GPDM	Gaussian Process Dynamical Model
GP-LVM	Gaussian Process Latent Variable Model
GTM	Generative Topographic Mapping
GTM-TT	Generative Topographic Mapping Through Time
HHM	Hidden Markov Model
ISOMAP	Isometric Feature Mapping
LE	Laplacian Eigenmaps
LLE	Local Linear Embedding
LVM	Latent Variable Model
MDS	Multidimensional Scaling
MF	Magnification Factor
ML	Machine Leaning
MTS	Multivariate Time Series
MVD	Multivariate Data
MVU	Maximum Variance Unfolding
NLDR	Non-linear Dimensionality Reduction
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PPCA	Probabilistic Principal Component Analysis
PR	Pattern Recognition
SML	Statistical Machine Leaning
SOM	Self Organizing Maps
t**GTM**	Student-t Generative Topographic Mapping
VB-GTM-TT	Variational Bayesian GTM Through Time

Chapter 2

Probabilistic Modelling

The increasing availability of high-dimensional data sets, with different levels of complexity and growing diversity of characteristics, sometimes under the fashionable remit of *Big Data*, is one of the driving forces behind recent advances in the development of machine learning techniques (c.f. Fig. 2.1 for examples of high-dimensional datasets).

Data from real-world processes and phenomena are likely to involve quality issues. They may include acquisition errors of several types, as well as the presence of noise. Within the constraints of this context, probabilistic modelling turns out to be a powerful approach due to its flexibility, and also one that is ideally suited to deal with uncertainty in its many forms.

In this chapter, we summarily review the theory of probabilistic modelling in the context of dimensionality reduction. We focus on models of the manifold learning family oriented towards exploratory multivariate data visualization. In particular, we present the details of some specific models, namely the generative topographic mapping (GTM) in section § 2.2 and the Gaussian process latent variable model (GP-LVM)in section § 2.3, both of which will be used in the following chapters to display experimental results.

2.1 Probabilistic dimensionality reduction

Non-linear dimensionality reduction (NLDR) methods [Lee and Verleysen, 2007] provide a flexible alternative for multivariate data modelling and representation that can lead to more faithful models of the observed data compared to those obtained with their simpler linear counterparts.

One specific approach is to perform probabilistic NLDR defining a model that

introduces a set of unobserved (or latent) variables $\mathbf{X}$ that can be related to the observed ones $\mathbf{Y}$, in order to define a joint distribution over both. These models are known as latent variable models (LVMs).

The latent space is dominated by a prior distribution $p(\mathbf{X})$ which induces a distribution over $\mathbf{Y}$ under the assumption of a probabilistic mapping of the form:

$$y_{i,j} = f_j(\mathbf{x}_i) + \epsilon_i, \tag{2.1}$$

where $\mathbf{x}_i \in \mathbb{R}^q$ is the latent point associated with the i^{th} observation $\mathbf{y}_i \in \mathbb{R}^p$, j is the index of the features of $\mathbf{Y}$, and ϵ_i is a noise term that accounts for noise in the data as well as for inaccuracies in the model. The noise is typically chosen to be Gaussian distributed $\epsilon \sim \mathcal{N}(0, \beta^{-1})$, where β is the precision.

One of the advantages of this approach is that it accommodates DR in an intuitive manner, if we assume that the dimensionality of the latent space is significantly lower than that of the observation space. In this case, the reduced dimensionality provides us with both implicit regularisation and a low-dimensional representation of the data, which can be used for visualisation (and, therefore, for data exploration [Vellido et al., 2011]) if the dimension is low enough. If the mapping f in Eq. (2.1) is taken to be linear and equal to a matrix $\mathbf{W} \in \mathbb{R}^{p \times q}$:

$$y_{i,j} = \mathbf{w}_j \mathbf{x}_i + \epsilon_i, \tag{2.2}$$

where $\mathbf{w}_j$ are the rows of $\mathbf{W}$, this model is known as probabilistic version of PCA [Roweis, 1997; Tipping and Bishop, 1999]. Given a Gaussian prior $p(\mathbf{X})$ over the latent variables, PCA is recovered in the limit as the precision β is going to infinity. The conditional probability of the data given the latent space can be written as

$$p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_i \mid \mathbf{W}\mathbf{x}_i, \beta^{-1}\mathbf{I}). \tag{2.3}$$

With a further assumption of independence across data points, the marginal likelihood of the data is

$$p(\mathbf{Y} \mid \mathbf{W}, \beta) = \int \prod_{i=1}^{N} p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{W}, \beta) p(\mathbf{x}_i) d\mathbf{X}. \tag{2.4}$$

It can be proven [Tipping and Bishop, 1999] that the maximum likelihood solution for $\mathbf{W}$ spans the principal sub-space of the data (even when the precision is finite).

In general, this approach can be applied to both linear and non-linear dimensionality reduction models, leading to the definition of, for instance, Factor Analysis [Bartholomew, 1987], GTM [Bishop et al., 1998a], or GP-LVM [Lawrence, 2005], to name just a few.

In the classic approach, the latent variables are marginalised (integrated out) and the parameters are optimised by maximising the model likelihood. An alternative (and equivalent) approach proposes to marginalise the parameters and optimise the latent variables, leading to the formulation of the GP-LVM model.

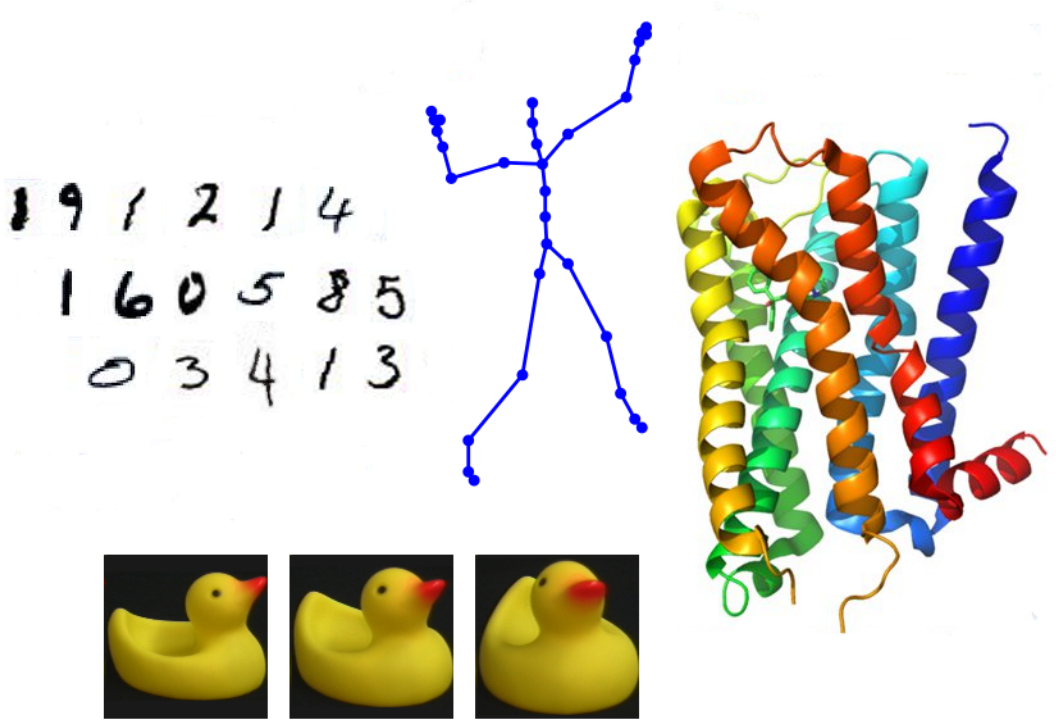


Fig. 2.1 Examples of high dimensional datasets of different nature. Top left: many samples of images of handwritten digits, from the MNIST dataset available at: http://yann.lecun.com/exdb/mnist/. Top center: sample pose of a human motion capture, from the CMU motion capture database available at http://mocap.cs.cmu.edu/. Right: image of a protein, taken from the home page of the Kappaim project https://sites.google.com/site/kappaaim/home. Bottom: image of an object captured during a rotation, from the Columbia Object Image Library (COIL) database available at http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php

2.2 Generative Topographic Mapping

The GTM is a non-linear LVM developed by Bishop, Svensén and Williams in the late nineties [Bishop et al., 1998a; Svensén, 1998]. In this model, manifold learning vector quantization techniques are applied and the observed variables are related with the latent ones through a non-linear function. The idea is to define a probability distribution in the latent space, in order to induce the corresponding probability distribution in the observed data space, using concepts of Bayesian inference.

GTM can be seen as a mixture of distributions whose centres are constrained to lay on an intrinsically low-dimensional space. Given that the generative model specifies a mapping from latent space to observed data space, such latent space can be used for data visualization when its dimensionality is equal to 1 or 2.

In the following section we first introduce the standard algorithm for GTM, § 2.2.1. Many extensions to the model have been proposed [Bishop et al., 1998b] over the years. For example, unless regularization is included, the GTM is prone to overfitting, and adaptive regularization for GTM was proposed in [Bishop et al., 1998b] and [Vellido et al., 2003]. Other extensions to GTM have been developed, such as those in [Vellido et al., 2006; Vellido, 2006b,a].

The GTM was redefined as a constrained Hidden Markov Model (HMM) by Bishop et al. [1997b] for the analysis of multivariate time series. The resulting GTM Through Time (GTM-TT), presented in section § 2.2.2 of this chapter, can be considered as a GTM model in which the latent states are linked by transition probabilities, in a similar fashion to HMMs.

2.2.1 The GTM model

Considering the noise model in Eq. (2.1), an example that generalises from the linear case to the non-linear one is the GTM, in which the mapping f is taken to be a linear combination of a set of M basis functions in the form

$$y_{i,j} = f_j(\mathbf{x}_i, \mathbf{W}) + \epsilon_i = \sum_{m=i}^{M} \mathbf{w}_j \phi_m(\mathbf{x}_i) + \epsilon_i. \tag{2.5}$$

In the current notation, $\mathbf{\Phi}$ is a set of M basis functions $\phi_m(\mathbf{x})$ (Gaussians in the standard model; other distributions can be considered for different types of data) and $\mathbf{W} \in \mathbb{R}^{p \times M}$ is a matrix of adaptive weight parameters $w_{i,j}$, where p is the dimension of the observed data space and $\mathbf{w}_j$ are the rows of $\mathbf{W}$.

This model, illustrated in Fig. 2.2, can be seen as a mixture of distributions whose

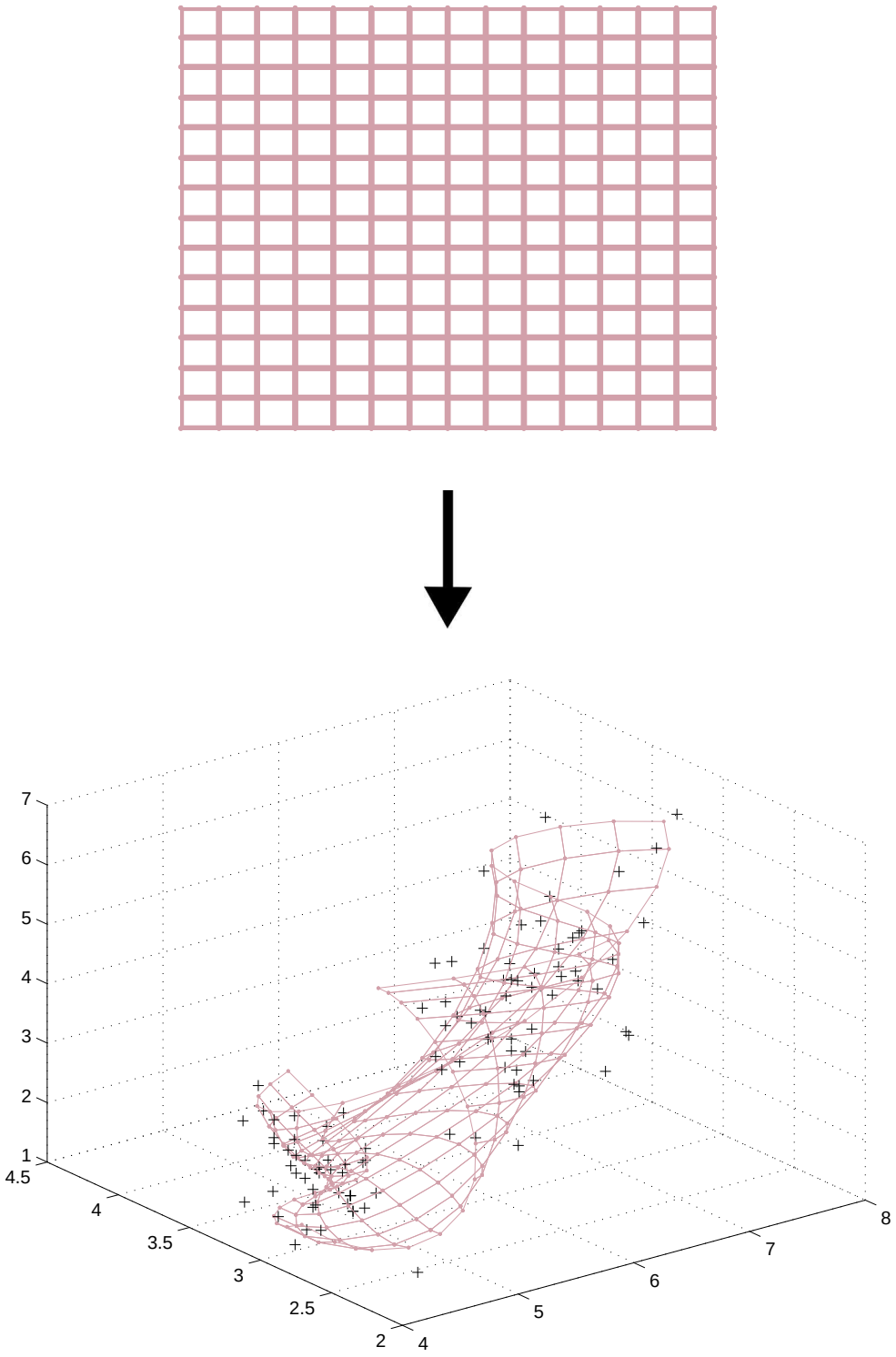


Fig. 2.2 Illustrative example of GTM mapping. A 15×15 2-D squared grid of latent points is mapped in a non-linear way onto a 3-D data space, fitting 2 clusters of random observations (black crosses). The 3-D plot shows how the support of the data can be represented with a 2-D surface embedded in an higher dimensional space.

centres $\boldsymbol{\mu}_m$ are constrained to lay on an intrinsically low-dimensional space. In the standard case, the basis distribution is chosen to be a set of Gaussian radial basis functions with the same lengthscale γ:

$$\phi_m(\mathbf{x}) = exp\left(-\frac{\gamma}{2} \parallel \mathbf{x} - \boldsymbol{\mu}_m \parallel^2\right), \tag{2.6}$$

but other distributions can be considered for different types of data; one example will be provided later on in § 4.4.1, in which a mixture if Student-t distributions is used to define a GTM variant that behaves robustly in the presence of atypical data or outliers.

The centres of the mixture of distributions can be interpreted as data prototypes or cluster centroids that can be further agglomerated in a full-blown clustering procedure. In this manner, GTM combines the functionalities of Self-Organising Maps (c.f. section §4.3) and mixture models by providing both data visualisation over the latent space and data clustering [Olier and Vellido, 2008c].

Provided a prior distribution over the latent space, this model leads, in a similar way to probabilistic PCA (PPCA) [Tipping and Bishop, 1999], to a Gaussian conditional distribution of the data

$$p(\mathbf{y}_i \mid \mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}\left(\mathbf{y} \middle| \sum_{m=i}^{M} \mathbf{w}_j^\top \phi_m(\mathbf{x}_i), \beta^{-1}\mathbf{I}\right) \tag{2.7}$$

$$= \left(\frac{\beta}{2\pi}\right)^{D/2} exp\left(-\frac{\beta}{2}\sum_{j=1}^{p}\left(y_{i,j} - \sum_{m=i}^{M} \mathbf{w}_j^\top \phi_m(\mathbf{x}_i)\right)^2\right). \tag{2.8}$$

Model likelihood and expectation maximization

The GTM algorithm aims to find the probability $p(\mathbf{y}|\mathbf{W}, \beta)$ of a data point given the adaptive weight parameters $\mathbf{W}$ and the noise variance β. To do so, the latent vectors $\mathbf{x}$ are integrated out of the model. To make the computation analytically tractable, the prior distribution $p(\mathbf{x})$ is defined by a set of K equally weighted delta functions

$$p(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} \delta(\mathbf{x} - \mathbf{x}_k). \tag{2.9}$$

The K centres $\mathbf{x}_k$ are distributed in a predefined regular lattice, which is taken to be squared in the standard case (other choices are allowed, for example hexagonal grids). This discrete choice of the prior distribution simplifies the integration and,

as a result, the data distribution becomes

$$p(\mathbf{y}|\mathbf{W},\beta) = \int p(\mathbf{y}|\mathbf{x},\mathbf{W},\beta)p(\mathbf{x})dx = \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{x}_k,\mathbf{W},\beta), \tag{2.10}$$

and assuming the data points i.i.d., we obtain the following final expression of the model likelihood:

$$\mathcal{L} = \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{W},\beta). \tag{2.11}$$

To estimate the parameters $\mathbf{W}$ and β, we can use a maximum likelihood approach, which is equivalent to consider the maximum of the log-likelihood

$$\ell = log(\mathcal{L}) = \sum_{n=1}^{N} log\left(\frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}_n|\mathbf{x}_k,\mathbf{W},\beta)\right). \tag{2.12}$$

The optimization of the adaptive parameters can be achieved by any standard non-linear optimization technique (see e.g. [Press et al., 1988]) but, since we are working with a mixture of Gaussians, the most common choice is to use the expectation-maximization (EM) algorithm [dem; Bishop, 1995]. Given the initial values for $\mathbf{W}$ and β, the E-step for the standard GTM formulation is the same as for the general Gaussian Mixture model, where the the conditional probability of each latent point given each observed data point is computed using Bayes' theorem. The probabilities are usually referred to as the responsibilities r_{kn}

$$r_{kn} \equiv p(\mathbf{x}_k|\mathbf{y}_n,\mathbf{W},\beta,) = \frac{p(\mathbf{y}_n|\mathbf{x}_k,\mathbf{W},\beta)p(\mathbf{x}_k)}{\sum_{k'=1}^{K} p(\mathbf{y}_n|\mathbf{x}_{k'},\mathbf{W},\beta)p(\mathbf{x}_{k'})}. \tag{2.13}$$

Notice that, for the choice of prior distribution made in Eq: (2.9), the effect of the term $p(\mathbf{x}_k)$ is cancelled. Considering now the choice of the GTM mapping made in Eq. (2.5), we obtain that the M-step of the EM algorithm reduces to the solution of a set of linear equations. For more details about the EM algorithm and the update equations for the standard GTM see [Bishop et al., 1998a, § 2.2].

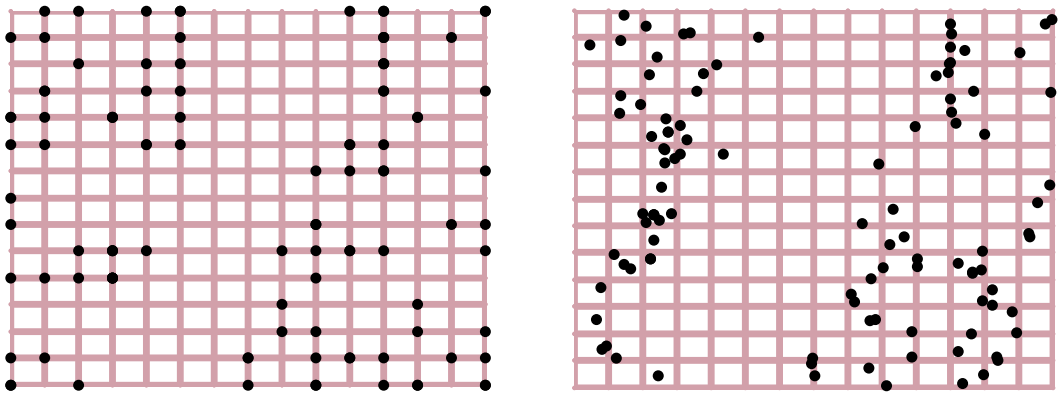


Fig. 2.3 Visualization of modes projections (black dots, left diagram) and means projections (black dots, right diagram) on the 2-D GTM latent space for the 3-D artificial dataset in Fig. 2.2.

Data Visualization

Since we are especially interested in data visualization, we can use the conditional probability defined by Eq: (2.13) to obtain both a posterior mode projection of $\mathbf{y}_n$

$$k_n^{mode} = \underset{k}{\texttt{argmax}}\, r_{k,n} \tag{2.14}$$

(which implies assigning each observed data point to that latent point with the highest responsibility for its generation), or a posterior mean projection

$$\mathbf{x}_n^{mean} = \sum_{k=1}^{K} r_{kn}\mathbf{x}_k \tag{2.15}$$

(locating the observed data point at a location in latent space that results from a responsibility-weighted combination of all latent point locations).

We can now visualize (see Fig. 2.3) the observed data points over the low-dimensional latent space using the posterior mean projection, which also provides an assignment of each data point to a representative cluster. For a visualisation purpose, the typical setting is a latent dimension $q = 2$ or 3.

2.2.2 Time-series analysis with GTM-TT

When the observed data space is known to be in the form of a time series, the time-dependent nature of the observations makes the assumption of i.i.d. inappropriate. In order to make the GTM model suitable to the analysis of temporal data, we consider here an extension within the framework of hidden Markov models (HMMs)

[Rabiner, 1989]. This model is known as GTM Through Time (GTM-TT) [Bishop et al., 1997b] and was proposed almost in parallel to the standard GTM.

The GTM-TT can be interpreted as a standard GTM model in which the latent points are considered as hidden states

$$\mathbf{Z} = \{\mathbf{z}_t\}_{t=1,\ldots,T}$$

for every time step t. Similarly to HMMs, the states are connected by a transition probability $a_{ij} = p(\mathbf{z}_j|\mathbf{z}_i)$, which represents the probability of making a transition to the state j from the current state i. We denote with $\mathbf{A}$ the matrix which describes the transition states

$$\mathbf{A} = \{a_{ij}\} : a_{ij} = p\left(\mathbf{z}_t = \mathbf{x}_j|\mathbf{z}_{t-1} = \mathbf{x}_i\right), \qquad i, j = 1, \ldots, K \tag{2.16}$$

where K is the number of allowed hidden states $\mathbf{x}_k$ (which is the number of vector prototypes). Given the initial state probabilities on each of the latent points at the first time step $t = 1$

$$\boldsymbol{\pi} = \{\pi_k\} : \pi_k = p\left(\mathbf{z}_1 = \mathbf{x}_k\right), \tag{2.17}$$

then the parameters governing the GTM-TT model are

$$\boldsymbol{\Theta} = (\boldsymbol{\pi}, \mathbf{A},\ \mathbf{W}, \beta)\,. \tag{2.18}$$

Notice that the parameters $\mathbf{W}$ and β, together with the transition probabilities $\mathbf{A}$, are common to all time steps of the GTM algorithm, so that the number of adaptive parameters in the model is independent of the length of the time series.

The adaptive parameters of the model can now be computed (in a similar way to GTM) using a maximum likelihood approach, via EM algorithm. In the context of HMMs, this is generally known as the Baum-Welch algorithm. Given an observed p-variate time series $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1,\ldots,p}$, the complete data log-likelihood is given by

$$\begin{aligned}\ell = &\sum_{k=1}^{K} v_{k,1} \log \pi_k + \sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{k=1}^{K} v_{i,n-1} v_{k,n} \log a_{ik} \\ &+ \frac{pN}{2} \log\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} v_{k,n} \parallel \mathbf{y}_n - \mathbf{f}(\mathbf{x}_k, \mathbf{W}, \beta) \parallel^2,\end{aligned} \tag{2.19}$$

where the binary vector $\boldsymbol{v}_n$ such that its component $v_{k,n}$ returns 1 if $\mathbf{z}_n$ is in state k, and zero otherwise (this indicators are suitable to simplify the expression of the

likelihood in order to perform the EM steps). Moreover, $\boldsymbol{v}_n$ satisfies $\sum_{k=1}^{K} v_{k,n} = 1$. A detailed description of the updating equations of the EM algorithm for GTM-TT can be found in [Bishop et al., 1997b, § 4].

Visualization of Time Series

In the same fashion as GTM, the GTM-TT allow data visualization simultaneously to data clustering. In this way we have a low-dimensional (usually $q = 2$) latent space where the multivariate time series is represented by the means of the posterior-mode projection, defined as

$$k_n^{(\texttt{mode})} = \underset{k}{\texttt{argmax}}\, r_{k,n} \tag{2.20}$$

where $r_{k,n}$ are the responsibilities probabilities

$$r_{k,n} \equiv p(\mathbf{z}_n = \mathbf{x}_k | \mathbf{y}_n, \boldsymbol{\Theta}) \tag{2.21}$$

This model, even if useful for MTS clustering and visualization, does not involve any regularization process.

2.3 Gaussian Processes Latent Variable Models

In this section we present the GP-LVM model, which is a Gaussian Process based dimensionality reduction model. To do so, in this section we firstly review the theory of Gaussian Processes (GPs) in section, including GPs for regression and some intuition about covariance functions. After this, we describe the details of the GP-LVM model.

2.3.1 Introduction to Gaussian Processes

A GP is used to describe distributions over functions and it is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution [Rasmussen and Williams, 2006].

Let the vector $\mathbf{x} \in \mathbb{R}^q$ and the function $f : \mathbb{R}^q \to \mathbb{R}$. A GP is a stochastic process determined by its mean function $\mu(\mathbf{x})$ and its covariance function $k(\mathbf{x}, \mathbf{x}')$, and it is denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{2.22}$$

The function f takes values over continuum on the input space (infinite input values). But, in practical applications, we only consider a finite set of instantiations of the function, since (computationally) we can only have access to a finite number of input vectors. If we consider a collection of inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1,\dots,N}$, we can generate a random vector of function values

$$\mathbf{f} = f(\mathbf{X}) = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)) \in \mathbb{R}^N \tag{2.23}$$

which is Gaussian distributed with covariance matrix $\mathbf{K}$ given by the gram matrix of the covariance function k, denoted with $\mathbf{K}_{\mathbf{f},\mathbf{f}}$

$$\mathbf{K}_{\mathbf{f},\mathbf{f}} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdot & \cdot & \cdot & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdot & \cdot & \cdot & k(\mathbf{x}_2, \mathbf{x}_N) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdot & \cdot & \cdot & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{2.24}$$

In this document we equivalently refer to the covariance function k as the kernel function or simply the kernel.

The kernel k defines the *correlation* between two inputs and can be interpreted as a similarity or as a distance into the functional space of positive semidefinite kernels k. There is, in fact, a relation between GP prediction and the regularization theory in reproducing kernel Hilbert spaces (RKHSs), as described in detail by Rasmussen and Williams [2006, § 6].

Without loss of generality, the mean function is typically chosen to be equal to zero.

An intuitive understanding of the distribution of $f(\mathbf{X})$ can be gained if we think of it as a marginal distribution. In fact, the input space $\mathcal{X}$ can be divided in two sets of input vectors, the observed ones $\mathbf{X}$ and the unobserved (potentially infinite) ones: due to the properties of joint Gaussian distributions, we can integrate over the unobserved variables. A sample of the function f can be obtained by sampling from its distribution following the standard procedure for Gaussians. (For more technical details about Gaussian distributions and mathematical identities check Appendix A).

Gaussian Processes for regression

Let's consider a regression problem in which a set of observations $\mathbf{X} = \{\mathbf{x}_n\}_{n=1,\dots,N}$ is mapped into a set of outputs $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1,\dots,N}$. We want to find the function f that maps the inputs into the outputs. Using Bayesian statistics, we can perform in-

ference over the function values by combining two pieces of information: our prior belief over the properties of f, encoded in the *prior* distribution, and the information given by the input data, encoded in the *likelihood* distribution. We assume that $\mathbf{Y}$ constitutes a noise-corrupted version of $f(\mathbf{X})$, according to Eq. 2.1. That is, we assume that $\mathbf{y}_n$ it is obtained by the correspondig $f(\mathbf{x}_n)$ with the addition of a Gaussian noise of β^{-1} variance. Then we have

$$p(\mathbf{f} \mid \mathbf{X}, \mathbf{Y}) = \frac{\overbrace{p(\mathbf{f} \mid \mathbf{X})}^{\text{prior}} \overbrace{p(\mathbf{Y} \mid \mathbf{X}, \mathbf{f})}^{\text{likelihood}}}{\underbrace{p(\mathbf{Y} \mid \mathbf{X})}_{\text{marginal likelihood}}} \tag{2.25}$$

Assuming i.i.d. inputs $\mathbf{X}$ and a likelihood following a Gaussian distribution, if we use a GP prior over the mapping f we have

$$p(\mathbf{f} \mid \mathbf{X}, \mathbf{Y}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}) \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n \mid \mathbf{f}, \beta \mathbf{I}). \tag{2.26}$$

The computation of the marginal likelihood has been solved analytically using the matrix determinant lemma and the Woodbury identity.

Prediction of an unobserved function value $\mathbf{f}_* = f(\mathbf{x}_*)$ computed in a test point $\mathbf{x}_*$ is obtained considering the joint distribution

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{f}_*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top & \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} \end{bmatrix}\right), \tag{2.27}$$

where the elements of the covariance matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ are given by 2.24 and $\mathbf{K}_{\mathbf{f}_*,\mathbf{f}}$ and $\mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*}$ are

$$\begin{aligned} \mathbf{K}_{\mathbf{f}_*,\mathbf{f}} &= \begin{bmatrix} k(\mathbf{x}_*, \mathbf{x}_1) \\ k(\mathbf{x}_*, \mathbf{x}_2) \\ \cdot \\ \cdot \\ k(\mathbf{x}_*, \mathbf{x}_N) \end{bmatrix} \\ \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} &= k(\mathbf{x}_*, \mathbf{x}_*) \end{aligned} \tag{2.28}$$

and, applying the properties of the conditional Gaussians (see Appendix A), it follows

$$p(\mathbf{f}_* \mid \mathbf{f}, \mathbf{X}) = \mathcal{N}(\underbrace{\mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{f}}_{\text{mean}}, \underbrace{\mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}}_{\text{covariance}}). \tag{2.29}$$

In applications to real datasets we are typically interested in prediction of noisy observations (as seen in Eq. 2.1). To do so, we can assume an additive i.i.d. Gaussian noise $\epsilon \sim \mathcal{N}(0, \beta^{-1})$ and we use a GP covariance $\tilde{\mathbf{K}} = \mathbf{K} + \beta^{-1}\mathbf{I}$. More details will be provided in section § 2.3.2 (in particularly Eq. 2.34) for GP-LVM.

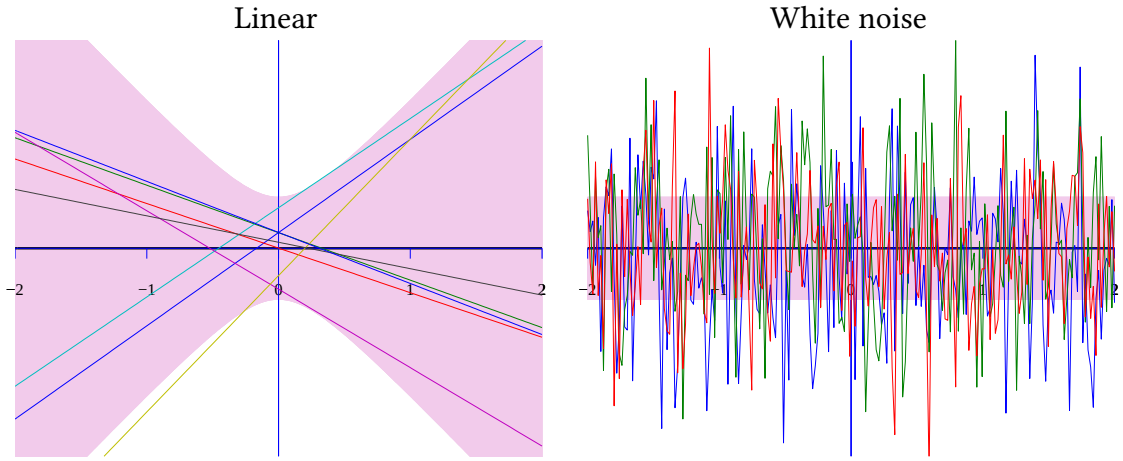


Fig. 2.4 Examples of different GP prior distributions over f according to different covariance functions: linear and noisy. Hear each function in different colors represents a different sample of f. The mean function is equal to *zero*, due to our choice of $m(\mathbf{x}) = \mathbf{0}$. The variance of f is here represented by the pink area (corresponding to the 95% confidence interval).

Covariance functions

In general, a GP prior is fully defined by its covariance. We present here the basic concepts about covariance functions needed for the understanding of the next chapters, and we refer to [Rasmussen and Williams, 2006, § 4] for a more detailed analysis.

Let's consider an input domain $\mathcal{X} \in \mathbb{R}^q$. A covariance function $k(\mathbf{x}, \mathbf{x}')$ (also known as kernel function) is a positive definite function which, intuitively, it is used to describe the similarity between two inputs.

A wide range of covariance functions can be used in GP models, and a correct choice might be crucial for specific applications. Each covariance function encodes, in a different way, the properties of the function we wish to learn. We can see in Fig. 2.4 and Fig.2.5 some examples of GP samples from different covariance functions: linear, white noise and periodic.

A widely used covariance function is the *exponentiated quadratic* (also known as

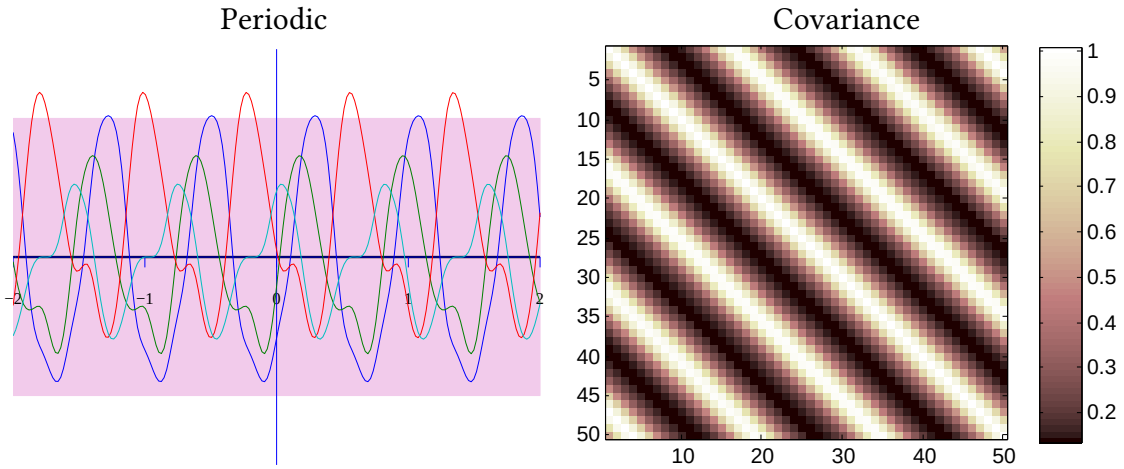


Fig. 2.5 Example of a Gaussian distribution of functions according to period kernel function. The corresponding covariance matrix is displayed on the right. We use 50 inputs equally distributed over the horizontal axis, with values taken between -1 and 2. High values of the covariance correspond to high correlation: notice that input points are correlated with a periodic structure (for example, the input x_1 is highly correlated with the inputs x_{17}, x_{33}).

squared exponential or *RBF* kernel):

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\omega}{2} \parallel \mathbf{x} - \mathbf{x}' \parallel_2^2\right). \tag{2.30}$$

The popularity of the exponentiated quadratic covariance function is due to the fact that it is capable of smoothly modelling different kind of functions only by varying the value of the lengthscale $1/\omega$. We can, in fact, find sensible initialization of the model by considering that the lenghtscale is proportional to the number of points where the function is crossing the *zero* axis.

In the left column of Fig. 2.6 we see different examples of $\mathcal{GP}$ priors over f. In the central and right columns of Fig. 2.6 we see the posterior distribution of f, after the observation of some data: notice that, far from the observed data, the distribution of the function tends to revert to the prior. The variance of the random variable is here represented by the pink area (corresponding to the 95% confidence interval).

2.3.2 Dimensionality reduction with GPs: the GP-LVM

Considering the mapping given by the noise model in Eq. (2.1), GPs have been used in probabilistic non linear dimensionality reduction to define a prior distribution over the mapping f, leading to the formulation of the Gaussian process latent variable model (GP-LVM).

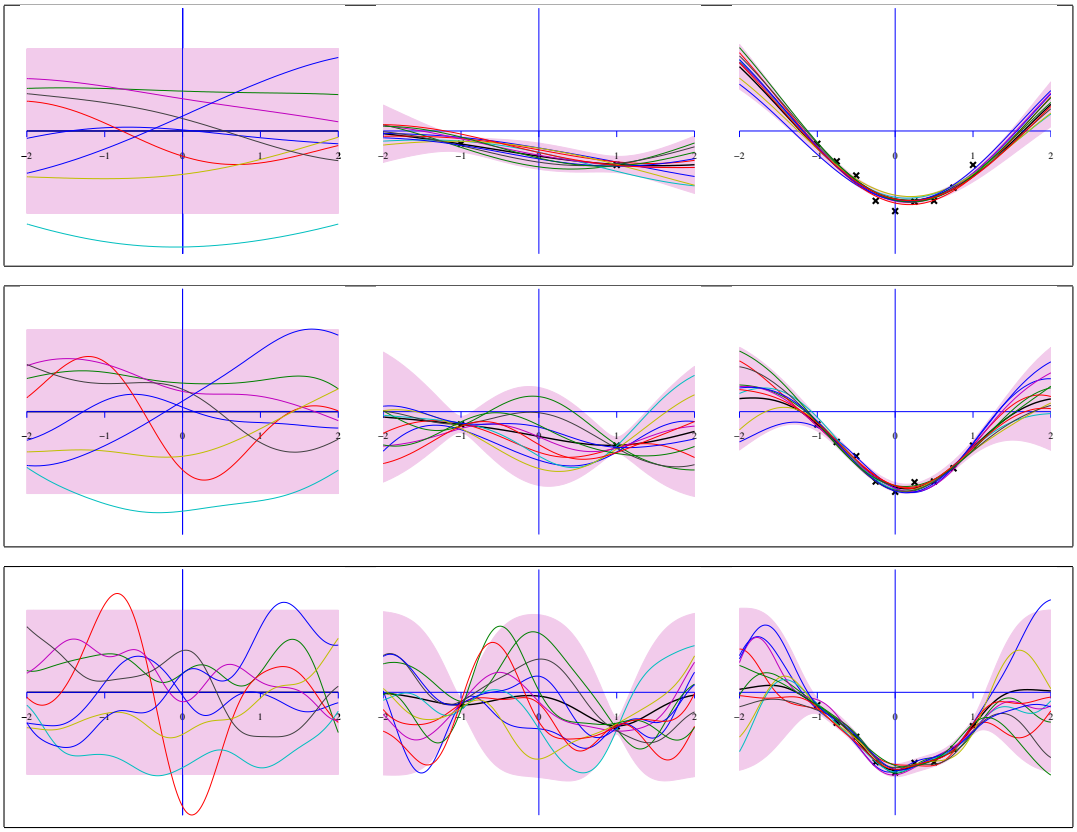


Fig. 2.6 This diagram shows, in each row, the effect of changing the lenghtscale in the *exponentiated quadratic* covariance function (from top to bottom: $\omega = 0.2, 1, 5$). In the first column we can see how the GP prior distribution is affected by this change. The second and third columns show some samples from the posterior after seeing 2 and 9 data points. Each function in different colors represents a different sample of f. The mean function is in black. The variance of f is here represented by the pink area (corresponding to the 95% confidence interval).

Lawrence [2005] introduces GP-LVM pointing out its dualistic relation with probabilistic principal component analysis (PPCA). The expression of the marginal likelihood of the PPCA model, as defined in Eq. 2.4, is a tractable integral which leads to the following solution

$$p(\mathbf{Y} \mid \mathbf{W}, \beta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n \mid \mathbf{0}, \mathbf{W}\mathbf{W}^\top + \beta^{-1}\mathbf{I}). \tag{2.31}$$

We now want to find the values of the parameters $\mathbf{W}$ which give the maximum value for the marginal likelihood of the data. As suggested in [Roweis, 1997; Tipping and Bishop, 1999], the solution of this problem can be treated as an eigenvalue problem, resulting in a fast computation.

In the standard dimensionality reduction approach described before, the likeli-

hood of the data $\mathbf{Y}$ given $\mathbf{X}$, as seen in Eq. (2.31) and Eq. (2.10), is computed by marginalising out the latent variables and optimising the mapping. In GP-LVM, on the contrary, the likelihood is computed by marginalizing out the linear mapping $\mathbf{W}$ and optimizing the latent variables $\mathbf{X}$, providing a dual expression of Eq. 2.31

$$p(\mathbf{Y} \mid \mathbf{X}, \beta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n \mid \mathbf{0}, \mathbf{X}\mathbf{X}^\top + \beta^{-1}\mathbf{I}). \tag{2.32}$$

However, since the mapping is integrated out, this means that we are free to consider non-linear mappings too. Indeed, the GP-LVM models considers non-linear mappings which are marginalised out after placing a GP prior on them. This leads to an expression of the likelihood which is similar to the one obtained for the GPs, with the difference of $\mathbf{X}$ being unobserved.

More specifically, here the likelihood is the product of p independent GPs, each one associated with a dimension of the observed dataset $\mathbf{Y}$, so that

$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{f}, \beta) = \prod_{j=1}^{p} \mathcal{N}(\mathbf{Y}_{:,j} \mid \mathbf{0}, \mathbf{K} + \beta^{-\mathbf{1}}\mathbf{I}) = \prod_{\mathbf{j=1}}^{\mathbf{p}} \mathcal{N}(\mathbf{Y}_{:\mathbf{j}} \mid \mathbf{0}, \tilde{\mathbf{K}}). \tag{2.33}$$

The GP-LVM prediction of an unobserved value $\mathbf{y}_*$ computed in a test point $\mathbf{x}_*$ is obtained considering the joint distribution

$$p(\mathbf{y}_* \mid \mathbf{Y}, \mathbf{X}, \mathbf{x}_*) = \prod_{j=1}^{p} \mathcal{N}(\mathbf{y}_* \mid \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{Y}_{:,j},\ \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{K}_{\mathbf{x},*}).$$

To follow the notation previously introduced, the noise model in Eq. (2.1) can be expressed as

$$y_{i,j} = \mathbf{K}_{\mathbf{f}_i,\mathbf{f}} \mathbf{K} \mathbf{Y}_{:,j} + \epsilon_i, \tag{2.34}$$

Derivatives of a GP

GPs have many interesting properties. We introduce here the analysis of the derivatives of the process, since this tool will be useful later on in this thesis for further investigation and application.

Due to the linear nature of the differential operator, the derivative of a GP is again a GP [Rasmussen and Williams, 2006, § 9.4], as long as the covariance function is differentiable. In fact, It GP allows to combine derivative information, and associated uncertainty with the function observations into the learning and inference process

[Solak et al., 2002].

This property allows inference and predictions about derivatives of a GP and, therefore, the Jacobian $\mathbf{J}$ of the GP-LVM mapping can be computed over continuum for every latent point $\mathbf{x}_*$ and we denote with $\frac{\partial \mathbf{y}_*}{\partial x^{(i)}}$ the partial derivative of $\mathbf{y}(x_*)$ with respect to the i^{th} component in the latent space:

$$\mathbf{J}^\top = \frac{\partial \mathbf{y}_*}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{y}_*}{\partial x^{(1)}}; \cdots ; \frac{\partial \mathbf{y}_*}{\partial x^{(q)}} \right], \tag{2.35}$$

where $\frac{\partial \mathbf{y}_*}{\partial \mathbf{x}}$ is a $q \times p$ matrix whose columns are multivariate normal distributions. We now consider the jointly Gaussian random variables

$$\begin{bmatrix} \mathbf{Y} \\ \frac{\partial \mathbf{y}_*}{\partial \mathbf{x}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}} & \partial \tilde{\mathbf{K}}_{\mathbf{f}_*,\mathbf{f}} \\ \partial \tilde{\mathbf{K}}_{\mathbf{f}_*,\mathbf{f}}^\top & \partial^2 \tilde{\mathbf{K}}_{\mathbf{f}_*,\mathbf{f}_*} \end{bmatrix} \right), \tag{2.36}$$

where $\partial \mathbf{K}_{\mathbf{f},\mathbf{f}_*}, \partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*}$ are a matrices given by

$$(\partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}})_{n,l} = \frac{\partial k(\mathbf{f}_n, \mathbf{f}_*)}{\partial x^{(l)}}, \quad \begin{matrix} n = 1, \cdots, N \\ l = 1, \cdots, q \end{matrix} \tag{2.37}$$

$$(\partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*})_{i,l} = \frac{\partial^2 k(\mathbf{x}_*, \mathbf{x}_*)}{\partial x^{(i)} \partial x^{(l)}}. \quad \begin{matrix} i = 1, \cdots, q \\ l = 1, \cdots, q \end{matrix} \tag{2.38}$$

The GP-LVM model provides an explicit mapping from the latent space to the observed space. This mapping defines the support of the observed data $\mathbf{Y}$ as a q dimensional manifold embedded into $\mathbb{R}^p$. If the covariance function of the model is continuous and differentiable, the Jacobian of the GP-LVM mapping is well-defined and the natural metric follows a Wishart Distribution (c.f. Chapter 6).

2.3.3 Illustrative example

High dimensional datasets coming from different domains have different levels of characteristics and raise different problems which we aim to solve. In computational biology, for example, we might want to use machine learning to design and interpret gene expression microarrays data, or we might want to infer the structure and characteristics of a protein given its amminoacid sequence. In computer vision machine learning can be used, for example, to model motions or videos, to infer missing sequences or to reconstruct partial images (whene occlusions or missing frames occur). Examples of datasets from different categories are given in Fig. 2.1.

To illustrate an example of dimensionality reduction, we describe here the hu-

man motion capture data from the *CMU Motion Capture Database*[1]. The dataset consists of a collection of human motions, taken from different subjects and acquired using 12 static cameras (more info about tool and data process are give here: http://mocap.cs.cmu.edu/info.php).

We show here a simple motion of jogging taken from the subject 35 of the dataset. We consider $N = 163$ input vectors and $p = 62$ features, corresponding the 3-D coordinates of 20 joints, plus the 2-D coordinates of the subject with respect to the floor. Example of 4 motion captures are displayed in Fig. 2.7.

Using the GP-LVM algorithm described in Sec. 2.3.2 we can learn a 2-D latent space and use it to visualise the motion captures, Fig. 2.8.

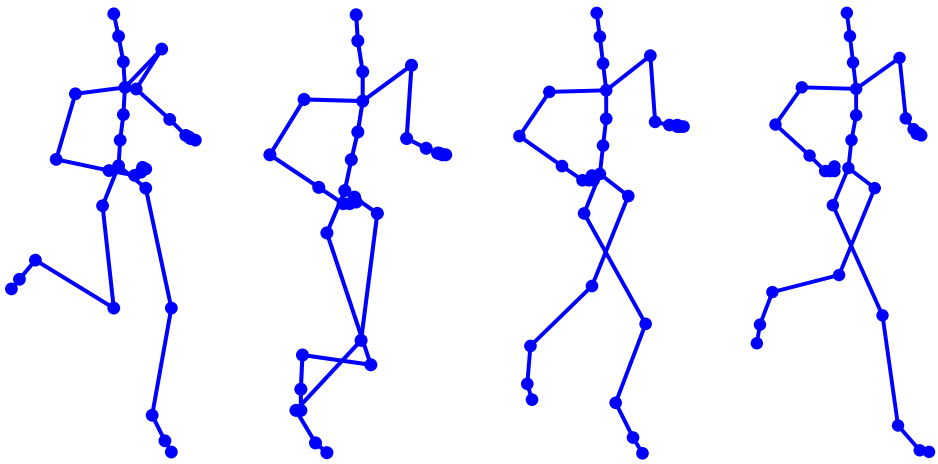


Fig. 2.7 Example of four poses of a jogging motion from the CMU motion capture database. Each motion capture is characterised by 20 joints (dots) connected by a skeleton.

[1]Public dataset, available at http://mocap.cs.cmu.edu/

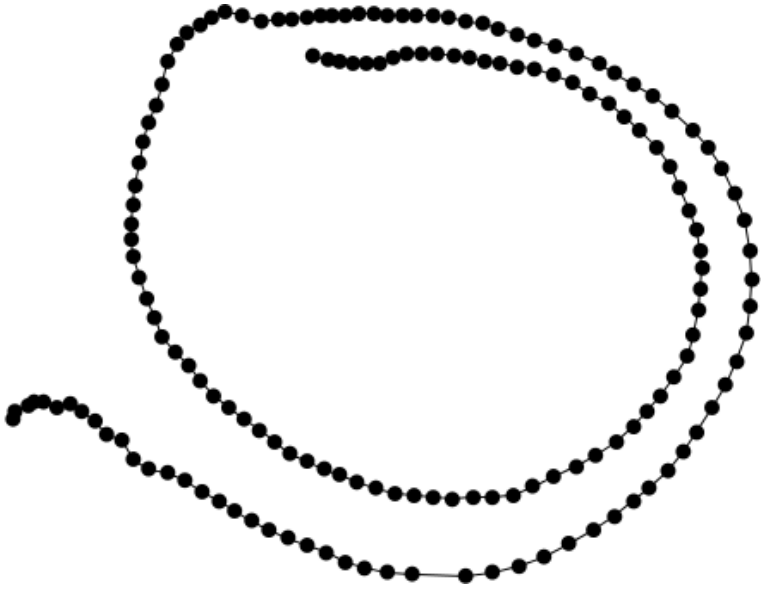


Fig. 2.8 Example of GP-LVM latent space for a jogging motion from CMU motion capture database. Each motion is project into a 2-D space and represented as a black dot. Notice that the low dimensional representation reflects the periodicity of a wealking motion.

Chapter 3

Some Tools for Improving Interpretability

Probabilistic modelling is at the heart of modern machine learning (in the form of statistical machine learning) This is the result of the confluence of two needs: the need to merge standard statistics with the modern algorithmic approaches to data analysis that machine learning provides and the need to deal with uncertainty in a principled way. The latter need responds, amongst other things, to the common presence of noise, in different forms, in the analysed data.

Even if these types of models are becoming increasingly popular, they are not without their practical problems. For instance, when combining probabilistic models with algorithms that involve the computation of distances or interpolants of the data space, our current tools are insufficient if they are defined according to the assumption that the observations span a Euclidean vector space, without taking into account the noise of the underlying geometry and the distortion produced by non-linear mappings.

In the first section of this chapter, we present a review of how the problem of local mapping distortion has been tackled in non-linear dimensionality reduction (NLDR). In the second section, we introduce the formalism of differential geometry that will be used later on in chapter § 6 as a tool needed to reinterpret distances and point-interpolations in a full probabilistic setting. In this way, we will be able to interpret the existing models in a meaningful way, thus making them more flexible and applicable even in those fields in which the interpretability as a key to usable knowledge extraction is as important as an optimal performance (such as, for instance, medicine, biology, finance, or astronomy, to name just a few).

As a solution to the fact that Euclidean geometry becomes insufficient when

dealing with noise and observations with complex structure and non linearities, the use of the tools of *Riemannian geometry* are becoming more popular among the machine learning community. In fact, we show that we can interpret an embedded Riemannian manifold as the underlying support of the data distribution. For every point on the manifold, all the geometrical properties are specified by a local positive definite matrix, called the *metric tensor*. Once this tensor is known, it is then possible to compute interesting objects as length minimizing curves (known as *geodesics*) and *magnification factors* (introduced in chapter § 3.3).

3.1 Dimensionality reduction and distortion measures

The use of dimensionality reduction techniques is an essential tool when dealing with visualization oriented applications involving high-dimensional data. Examples of popular models are Principal Component Analysis (PCA) and multidimensional scaling (MDS). PCA, independently introduced by Pearson [1901] and Hotelling [1933], provides a low dimensional linear representation of the data set which captures the most variation in the observed high-dimensional variables, while MDS [Mardia et al., 1979] aims to preserve distances and proximities between pairs of observations by the definition of a similarity matrix.

These methods are easy to interpret for practical purposes, but they also suffer from some limitations. For example, PCA is very useful for reducing redundancy of features in the original data set, but is restricted to the assumption of linearity, and MDS relies on the preservation of local distances, which is not always the best approach. The need to find less constrained (more flexible) methods for multivariate data modelling has led many researchers to explore and define non-linear techniques of dimensionality reduction (NLDR), which are becoming increasingly popular [Lee and Verleysen, 2007].

The most interesting contributions to this area range from spectral-based methods to manifold learning techniques. Some examples of spectral approaches include kernel PCA [Schölkopf et al., 1997], local linear embedding (LLE) [Roweis and Saul, 2000], isometric feature mapping (ISOMAP) [Tenenbaum et al., 2000], Laplacian eigenmaps (LE) [Belkin and Niyogi, 2003], or maximum variance unfolding (MVU) [Weinberger and Saul, 2006].

Advantages of spectral approaches include the reach of a global optimum and a smooth mapping from the data space to the low-dimensional space. On the other

side, when considering a probabilistic framework, the advantages typically include the explicit access to a (smooth) mapping from the low dimensional space to the high dimensional observed space. Other relevant advantages include marginalization of missing data, model selection through Bayesian inference and integration with other models (such as mixture models or temporal models).

Specific attention will be paid, in the following sections, to the analysis of probabilistic generative models of the latent variable models family. In particular, we will focus part of our research on models which can be expressed by the mapping given in Eq. (2.1). These models include the aforementioned models in § 2.1.

In general, NLDR techniques attempt to minimize the unavoidable distortion that they introduce in the mapping of the high-dimensional data from the observed space onto lower-dimensional spaces. For a more faithful interpretation of models, a large number of distortion measures have been proposed and adapted to visualization techniques for different NLDR methods.

While reducing dimensionality, NLDR generate different levels of local mapping distortion, that lead to a loss of information that we aim to recover, to some extent, into the visualization space. Stretching or compressing a space affects the preservation of pairwise distances between points. An example of such non-linear mapping is displayed in Fig. 3.1.

An interesting contribution comes from research by Aupetit [2007], who identifies different types of distortion, classified as geometrical and topological (including: manifold compression, stretching, gluing and tearing) and proposes the use of Voronoi diagrams and colour scales to visualize manifold-based measures such as point-based, segment-based and triangle-based measures.

The non-linearity of dimensionality reduction methods such as the Generative Topographic Mapping (GTM) and the Gaussian Process Latent Variable Model (GP-LVM) entails the existence of local distortion in the mapping of the data from the observed space onto the visualization space. This fact limits the direct interpretation of the visual data representation and there have been efforts to provide visual solutions to this limitation by defining and visualizing DR quality measures that, embedded in the method, can be associated to each data point, using colouring procedures for the data-corresponding cells in the Voronoi tessellation of the projection space.

A widespread used method for Self-Organising Maps (SOM) (which is not a true latent or generative model), called Unified distance Matrix (U-matrix), proposed in [Ultsch, 2003], allows the visualization of the pairwise distances between corresponding points in the original data space on the pseudo-latent space of the model.

The values of these distances can be represented with a color map accompanying the SOM topographic grid.

Another interesting approach, proposed in [Bishop et al., 1997a] for GTM (and extended to SOM), is the calculation of a magnification measure in a continuous way over the representation map. We delve into this technique in §3.3.

We show in the following section that, while dealing with generative LVMs where the mapping is continuous and differentiable, the expression of magnification factors (MFs) can be analytically derived. To do so, we will will use the tools of Riemannian geometry in order to formalize this approach.

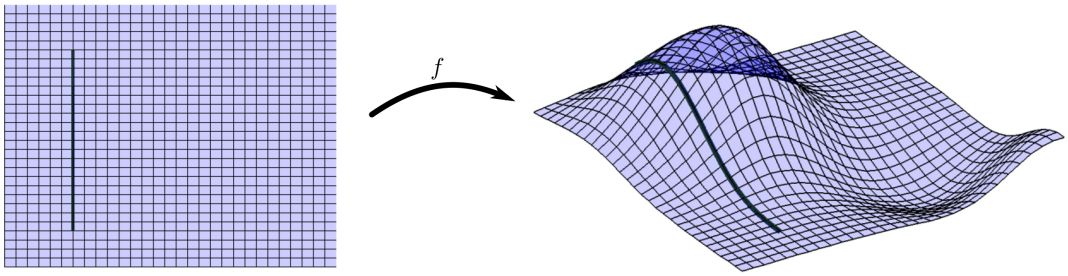


Fig. 3.1 Mapping between a 2-D plane and a surface embedded in a 3-D space. A straight line is subject to distortion under the non-linear projection.

3.2 Concepts of Riemannian Geometry

We study latent variable models as embeddings of uncertain surfaces (or manifolds) into the observation space. From a machine learning point of view, we can interpret this embedded manifold as the underlying support of the data distribution. To this end, we review the basic ideas of differential geometry, which study surfaces through local linear models.

Gauss' study [1827] of curved surfaces are among the first examples of (deterministic) latent variable models. He noted that a 2-dimensional surface $\mathcal{M}$ embedded in a 3-dimensional Euclidean space is well-described through a collection of mappings $\mathbf{x}_\alpha$ for each point $\boldsymbol{\xi} \in \mathcal{S}$

$$\mathbf{x}_\alpha : U \in \mathbb{R}^2 \rightarrow V \cap \mathcal{S} \in \mathbb{R}^3 \tag{3.1}$$

$$\alpha \mapsto \boldsymbol{\xi} \tag{3.2}$$

Intuitively, the regular surface is defined by a collection of open sets of $\mathbb{R}^2$ (small

pieces of planes) stick together in a way that the transition from a set to an other is made in a smooth way (with no sharp points, no self-intersections and no edges).

Historically, Gauss considered the case of two-dimensional surfaces embedded in $\mathbb{R}^3$, while the extension to higher dimensional *manifolds* is due to his student Bernhard Riemann [1854].

The pair $(\mathbf{x}_\alpha, U_\alpha)$ is called *parametrization* (or *system of coordinates*) on the manifold. A *differential manifold* is characterized by mappings $\mathbf{x}_\alpha$ that are smoothly varying (i.e., varying in a differentiable way) between the open sets U_α.

Given a smooth manifold, we can take advantage of its differential structure in order to make computations wit its elements. We can, in fact, define a smoothly-varying inner product on the tangent space $T\mathcal{M}$: this is a Riemannian metric.

Definition (Riemannian Metric): A Riemannian metric[1] $\mathbf{G}_\mathbf{x}$ on the differential manifold $\mathcal{M}$ is symmetric and positive definite matrix associated with a smoothly varying inner product $\langle \cdot, \cdot \rangle_\mathbf{x}$

$$\langle \mathbf{a}, \mathbf{b} \rangle_\mathbf{x} = \mathbf{a}^\top \mathbf{G}_\mathbf{x} \mathbf{b} \tag{3.3}$$

on the tangent space $T_\mathbf{x}\mathcal{M}$, for each point $\mathbf{x} \in \mathcal{M}$ and $\mathbf{a}, \mathbf{b} \in T_\mathbf{x}\mathcal{M}$. The matrix $\mathbf{G}$ is called the *metric tensor*. The pair $(\mathcal{M}, \langle \cdot, \cdot \rangle_\mathbf{x})$ is called *Riemannian manifold*.

Example 3.2.1: The most common and fundamental example of Riemannian manifold is the Euclidean space equipped with the canonical inner product $(\mathbb{R}, can)$.

Example 3.2.2: Suppose that we have an embedding $f : \mathcal{M} \longrightarrow N$, and that $(\mathcal{N}, \langle \cdot, \cdot \rangle_\mathbf{y})$ is a Riemannian manifold. We can construct a Riemannian metric on $\mathcal{M}$ by pulling back $\langle \cdot, \cdot \rangle_\mathbf{y}$ to $\langle \cdot, \cdot \rangle_\mathbf{x} = f * \langle \cdot, \cdot \rangle_\mathbf{y}$ on $\mathcal{M}$. In other words we have: $\mathbf{a}^\top \mathbf{G}_\mathbf{x} \mathbf{b} = \langle \mathbf{a}, \mathbf{b} \rangle_\mathbf{x} = \langle \frac{\partial}{\partial x} f(\mathbf{a}), \frac{\partial}{\partial x} f(\mathbf{b}) \rangle_\mathbf{y}$ where

$$\mathbf{G} = \mathbf{J}^\top \mathbf{J} \tag{3.4}$$

Remark: The Riemannian metric needs not be restricted to $\mathbf{G} = \mathbf{J}^\top \mathbf{J}$ and can be any smoothly changing symmetric positive definite matrix [do Carmo, 1992]. In this research we restrict ourselves to the more simple definition of Eq. 3.4 as it suffices for our purposes, but note that the more general approach has been used in machine learning, e.g. in *metric learning* [Hauberg et al., 2012] and *information geometry* [Amari and Nagaoka, 2000].

[1] For simplicity we use $\mathbf{x}$ instead of $\mathbf{x}_\alpha(\alpha)$ to denote a point on the manifold. The subscript $\mathbf{x}$ in $\mathbf{G}_x$ is omitted when there is no ambiguity.

The Riemannian metric encodes the geometrical properties of the manifolds and can be employed to compute useful quantities, such as curvatures, arc lengths and angles. In particular, we are interested in computing distances.

Definition (Geodesic Curve): Given two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}$, a *geodesic* is a length-minimising curve connecting the points

$$\gamma_g = \arg\min_{\gamma} \text{Length}(\gamma), \quad \gamma(0) = \mathbf{x}_1, \gamma(1) = \mathbf{x}_2, \tag{3.5}$$

where the length of a general curve $\gamma : [0, 1] \to \mathbb{R}^q$ is computed as

$$\text{Length}\,(\gamma) = \int_0^1 \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)}}.dt \tag{3.6}$$

It can be shown [do Carmo, 1992] that geodesics satisfy the following second order ordinary differential equation

$$\gamma'' = -\frac{1}{2}\mathbf{G}^{-1}\left[\frac{\partial \,\text{vec}\,\mathbf{G}}{\partial \gamma}\right]^{\top}(\gamma' \otimes \gamma'), \tag{3.7}$$

where $\text{vec}\,\mathbf{G}$ stacks the columns of $\mathbf{G}$ and $\otimes$ denotes the Kronecker product. The Picard-Lindelöf theorem [Tenenbaum and Pollard, 1963] then implies that geodesics exist and are locally unique given a starting point and an initial velocity.

3.3 Magnification Factors

The metric tensor defines the local geometrical properties of the considered generative LVM model and it can be used as a tool to data exploration. One way to visualise the tensor metric is through the differential volume of the high dimensional parallelepiped spanned by the mapping; this, for a latent dimension $q = 2$ is known as MF and it was introduced by Bishop et al. [1997a] for GTM (and standard SOM). Its explicit formulation, using the notation given in Eq. 3.4, is given by

$$\text{MF} = \sqrt{\det\left(\mathbf{J}^{\top}\mathbf{J}\right)}. \tag{3.8}$$

The MF points out the non-linear distortion generated by the projection of the observed data onto the representation map.

Being computed over continuum on the input space, the MF values can be visualised as a colormap over the 2-D display of the latent space of the chosen model. We

display in Fig. 3.2 a colormap representing MF values compute over a fine regular grid over the 2-D latent space of a GP-LVM model. The explicit formulation of the MF for the GP-LVM model is detailed later on in this document, in section §6.2.

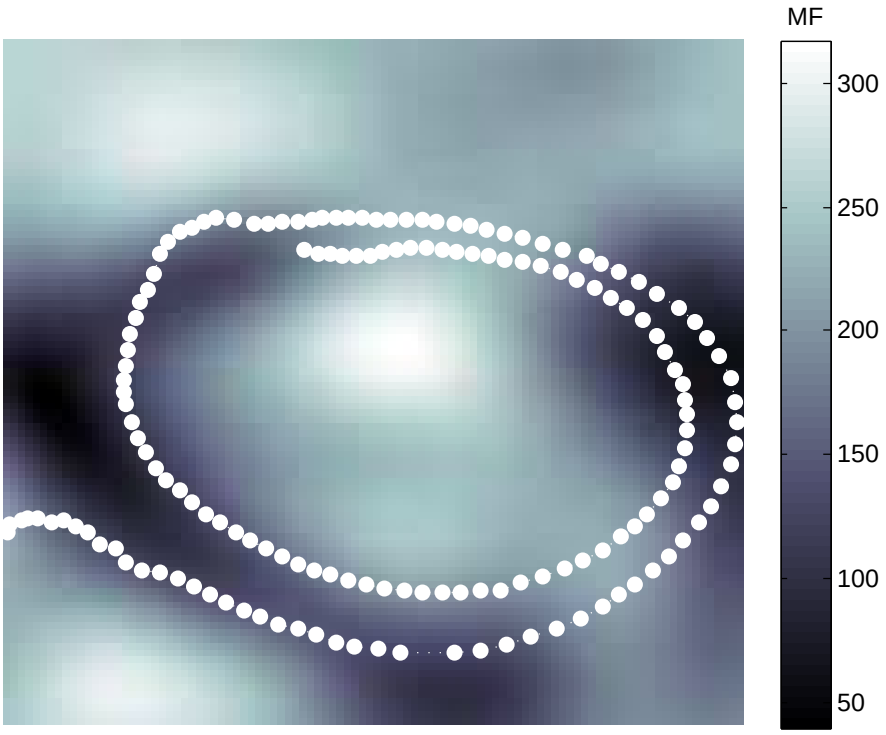


Fig. 3.2 Magnification Factor colormap on the GP-LVM latent space for a jogging motion from the CMU motion capture database.

Other examples of this type of this visualisation are provided throughout this document in the sections describing the experimental results. See Fig. 4.5 (top left), Fig. 4.6 (center), Fig. 4.7 (center), Fig. 5.1 (bottom right), Fig. 5.2 (bottom right), Fig. 6.1, Fig. 6.3 and Fig. 6.9.

We also introduce a 3-D plot of the MF colormap, in which the vertical axes represents the values of the MF. In Chapter 6 we define a distribution over the local metric tensor for generative latent variable models and in this context the 3-D display is used to visualise the variation of MF values between different samples from the metric, as in Fig. 7.1.

Another interesting example of visualization is provided in [Svensén, 1998, § 4.5], where the author visualizes the different levels of magnitude and directions of stretch of the MF over the GTM latent space.

The concept of MF has its origin in the field of computational neuroscience, where it evaluates the mapping distortion between the spatial density of biological

sensors and the two-dimensional spatial density of the corresponding topographic maps in the visual and somatosensory areas of the cortex. More specifically, the cortical MF would indicate the linear distance along the primary visual cortex concerned with each degree of visual field [Pointer, 1986], although controversy remains on whether the cortical magnification of the central visual field reflects its selective amplification, or merely reflects the ganglion cell density of the retina [Wässle et al., 1990]. As expressed in the context of vector quantization models [Hammer et al., 2007], local magnification is the result of a specific connection of the density of model prototypes and stimuli.

One of the most interesting facts is that the distortion caused by the non-linear mapping can be explicitly quantified as a MF over the latent space used for data visualization. For this property, the concept of magnification has recently been applied to manifold learning methods for NLDR, in order to visualize the distortion due to the embedding of a manifold in a high-dimensional space [Tosi and Vellido, 2012; Vellido et al., 2013; Tosi and Vellido, 2013; Gianniotis, 2013; Tosi et al., 2014b,a]. More details about novel visualization techniques involving MF are presented in the following chapter.

Chapter 4

Advances in Mapping Distortion Visualization for Non-linear Dimensionality Reduction Using Cartograms

One of the main advantages of non-linear dimensionality reduction methods, as mentioned in previous chapters, is their flexibility in the process of multivariate data modelling. Unfortunately, this flexibility is also accompanied by limitations, such as the difficult interpretation of the data (visual) representations they generate. Even latent variable manifold learning models, which represent multivariate data in low-dimensional representation spaces, can be difficult to make sense of, due to the fact that their coordinates in latent space are complex non-linear transformations of the observed ones, so that heterogeneous levels of local distortion are generated. These locally varying distortions and the loss of straightforward meaning in the low dimensional variables make tools to assist their interpretation an almost compulsory requirement.

Linear dimensionality reduction methods, on the other hand, are less flexible and their data representation can be less faithful as a result, but they compensate for this with the straightforward interpretation of their representation coordinates. This comes a long way to explain the popularity of linear dimensionality reduction techniques and the difficulties for non-linear ones to become mainstream.

Given that some of the modelling techniques considered in this thesis are of non-linear nature, we are faced with an interpretability challenge. In the current chapter, we respond to this challenge using a technique for data visualization in non-

linear latent models that is inspired in geographical representation methods. This technique is suited to both static data and multivariate time series representation.

The technique in question, known as Cartogram, draws inspiration from both geographic representations and physics. We introduce novel variants of Cartogram-based visualization for NLDR techniques. We illustrate the proposed method providing a self-contained description of the algorithms for the visualization of the Batch-SOM algorithm in § 4.3 and the robust tGTM algorithm in § 4.4.

4.1 Cartograms

In the area of thematic representations for geography, there is a particular type of mapping known as Cartogram. In this mapping, specific areas, often delimited by political borders, are locally distorted (both stretched or compressed) to convey the information on locally-varying underlying quantities of interest such as population density or socio-economic data. Examples of applications of Cartogram techniques include the visualization of census data, disease incidence, birth rate, and annual income.

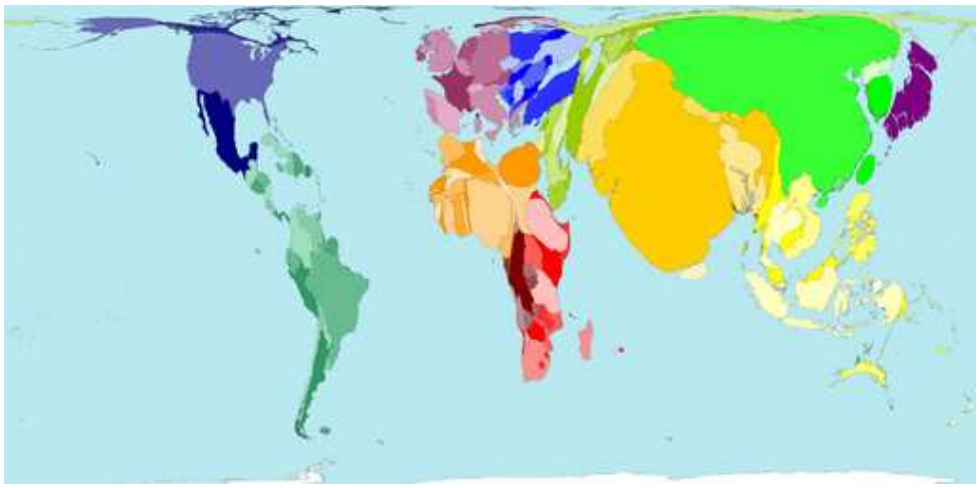


Fig. 4.1 Cartogram representation of the world map according to the population density: countries with a higher population density are represented with an area that is bigger than the geographical one. In this way it is easy to visually notice the contrast between countries with high population densities (such as India or Japan) and countries with lower population density (such as Australia or Canada). Source of image: http://www.worldmapper.org/display.php?selected=2.

In the last decades, the development of feasible computer-based Cartograms has been a challenging undertaking and several methods have been proposed to accomplish it. An extended review of the recent history of this subject is given by

Tobler [2004], where the author completes and unifies pioneering work started in the 1960's. Nowadays, the use of cartograms for the visual representation of socio-economic data in geographical maps has become popular thanks to public resources such as Worldmapper[1] (An example is provided in Fig. 4.1).

From a mathematical point of view, the geometrical distortion of Cartograms takes the form of a continuous transformation from $\mathbb{R}^2$ to itself. Fig. 4.2 represents the distortion of a square patch on the original plane: here a vector $\mathbf{x} = (x^{(1)}, x^{(2)})$ is mapped onto a vector $\mathbf{x}' = (x'^{(1)}, x'^{(2)})$ according to a transformation $\mathcal{T}$ in such a way that the Jacobian of the transformation is proportional to an underlying *distortion* variable d, which describes the *deformation* of the patch quantitatively :

$$\begin{aligned} \mathcal{T}: \quad & \mathbb{R}^2 \to \mathbb{R}^2 \\ & \mathbf{x} \to \mathcal{T}(\mathbf{x}) = \mathbf{x}' \end{aligned}$$

$$\left[\frac{\partial \mathcal{T}_i}{\partial x^{(j)}}\right]_{i,j} \propto d. \qquad (4.1)$$

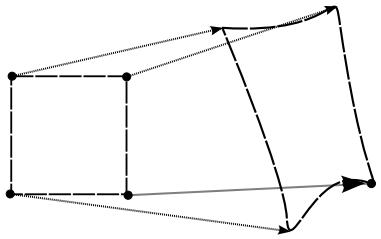


Fig. 4.2 A continuous transformation is applied to a 2-D square, which is mapped into a distorted patch (dashed line).

The Cartogram projection is not determined uniquely by the condition 4.1, since the problem has two degrees of freedom. To fix the projection we need one more constraint and, to do so, we can operate in many different ways. Since there is not a single choice for the constraints of the projection $\mathcal{T}$, some choices may result in loss of connectivity between the fragment borders or overlapping of neighbours areas.

A method for building Cartograms, based on the physics' principle of linear diffusion processes, has been proposed by Gastner and Newman [2004]. The principle of diffusion applies the theory of parabolic partial differential equations to the problem of diffusion of a large number of particles, knowing their density as a function of the position and the time. In a natural system, the particles will flow, over time, from areas of higher density to areas of lower density, resulting in a final state where the overall density is homogeneous. In order to apply this model to the problem of Cartograms, the distorting variable d is interpreted as a diffusion function $\rho(\mathbf{x}, t)$ depending on the position $\mathbf{x} \in \mathbb{R}^2$ and time $t \in [0, \infty)$. The diffusion function is

[1] www.worldmapper.org

normalized over the map $d = \frac{\rho(\mathbf{x},t)}{\bar{\rho}}$, where $\bar{\rho}$ is the average population density over the area to be distorted. Intuitively, we start with a map where different areas are characterized by different density of the chosen variable (population density, as an example) and we let the original locations diffuse for $t \to \infty$. The resulting map presents a consistent distortion: every point has been displaced in order to have the same density (as displayed in Fig. 4.1). For this reason, Cartograms are also known as *density-equalizing maps*, as in Gastner and Newman [2004]. The system can be associated to the partial differential equation

$$\nabla^2 \rho - \frac{\partial \rho}{\partial t} = 0, \tag{4.2}$$

which has to be solved for $\rho(\mathbf{x}, t)$, assuming that the initial condition corresponds to each map fragment being assigned its value of ρ. The distortion diffusion velocity is defined as $\mathbf{v}(\mathbf{x}, t) = -\frac{\nabla \rho}{\bar{\rho}}$ and, from it, the map location displacement can be calculated as

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{v}(\mathbf{x}, t')dt'. \tag{4.3}$$

As a result, the Cartogram is generated using the new locations.

To avoid arbitrary diffusion through the external map boundaries, the map is assumed to be surrounded by an area in which the distortion is set to be the mean distortion of the complete map. This guarantees that the total map area will remain constant. Moreover, the whole system, including the surrounded areas, is considered to be enclosed in a rectangular *box*, for simplicity. For further details about boundary conditions on the walls of the box and other calculations we refer to [Gastner and Newman, 2004].

4.2 Cartogram representation for NLDR methods

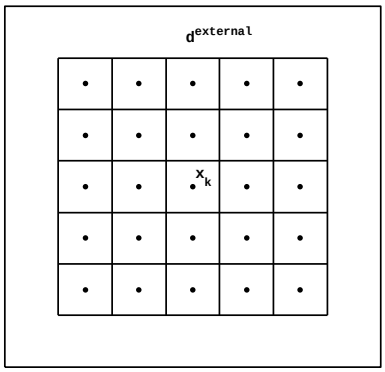
In this chapter, we describe and assess Cartogram representation as a tool for increasing the interpretability and usability of multivariate data visualization through non-linear dimensionality reduction methods (NLDR), extending recent work reported in [García et al., 2013]. We introduce the cartography-inspired method of Cartogram representation of mapping distortion in a way that should help to intuitively interpret the data visualization of the considered non-linear models.

As described in § 3.3, the distortion of non-linear models can be quantified over a continuum in the representation map in the form of an MF. The visualization of the

2-D latent space of the NLDR manifold learning methods that we use is transformed into a Cartogram taking in account these two points:

- A square regular grid generated by discretizing the latent space continuum forming a lattice centred in the points $\mathbf{x}_k$, $k = 1, \ldots K$ is used to define the map internal boundaries.
- It is assumed that the level of distortion $d^{external}$ in the space beyond this square is uniform and equal to the mean distortion over the complete map, that is

$$d^{external} \equiv \frac{1}{K} \sum_{k=1}^{K} MF_{\mathbf{x}_k},$$



where $MF_{\mathbf{x}_k}$ is the MF computed in $\mathbf{x}_k$. Likewise, we assume that the level of distortion within each of the squares (hexagons) associated to $\mathbf{x}_k$ is itself uniform.

An advantage of this Cartogram-based method is its *portability*: it is easy to implement for different representation architectures and alternative NLDR visualization techniques for which distortion can be quantified.

4.3 Cartogram representations for batch-SOM

4.3.1 Self-Organization Maps and their variants

A well-known and widely used NLDR vector quantization method for data visualization in low-dimensional spaces is Kohonen's SOM [Kohonen, 2001], in its many variants. This method attempts to model data through a discrete version of a low-dimensional manifold consisting of a topologically ordered grid of prototypes.

SOM simultaneously performs a combination of vector quantization and topographic representation and can be intuitively interpreted as a kind of nonlinear but

discrete PCA. The popular K-Means algorithm can also be seen as an specific instantiation of SOM. Its nonlinearity has not prevented SOM to achieve mainstream status, even in very practical application fields.

Let $\mathbf{Y}$ be a design matrix with N samples $\mathbf{y}_n$ of dimension p, according to the notation in section § 1.5. A SOM consists of a discrete layer (map) of prototypes, also called units or neurons due to its connectionist origins. The prototypes are arranged in a low dimensional regular grid of dimension q, which is often taken to be 2-D for ease of visualization.

The low dimensional prior structure consists of a set of units $\mathbf{x}_k \in \mathbb{R}^q$, $k = 1, \ldots, K$, which is related, through an embedding function, to the prototypes $\mathbf{m}_k \in \mathbb{R}^p$, $k = 1, \ldots, K$, in the data space.

The iterative algorithm initializes the weight vectors $\mathbf{m}_k$ randomly, chosen from the dataset $\mathbf{Y}$. Other types of initialization are used as well, for example according to some basic pre-projection of the data. For each data sample $\mathbf{y}_j$, $j = 1, \ldots, N$, it finds the best matching unit (BMU) $\mathbf{m}_{k_j}$ of index k_j, computed as

$$k_j = \underset{k}{\texttt{argmin}} \left\{ d(\mathbf{y}_j, \mathbf{m}_k) \right\},$$

where the distance function $d(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is usually defined as the Euclidean distance

$$L_2(\mathbf{y}, \mathbf{m}_k) = \|\mathbf{y} - \mathbf{m}_k\|,$$

although L_1 or L_∞ distances, for instance, can also be considered (c.f. [Kohonen, 2001]).

The update of the adaptive parameters is not limited to the BMU. This is not a *winner-takes-all* but a *winner-takes-most* algorithm. The weights of the BMU are the most modified, but neighbouring SOM units also undergo modification. This practically implemented by defining a neighbourhood function $h(\cdot, \cdot)$ that controls the range of the modifications. Different functions can be considered, such as

$$h(\mathbf{x}_k, \mathbf{x}_c) = exp\left(-\frac{1}{2\sigma^2} \parallel \mathbf{x}_k, \mathbf{x}_c \parallel^2 \right) \quad (Gaussian)$$

$$h(\mathbf{x}_k, \mathbf{x}_c) = \begin{cases} 0 & if\ d(\mathbf{x}_k, \mathbf{x}_c) > \lambda \\ 1 & if\ d(\mathbf{x}_k, \mathbf{x}_c) \leq \lambda \end{cases} \quad (bubble)$$

The prototype vector $\mathbf{m}_k$ is updated according the following rule:

$$\mathbf{m}_k^{(t+1)} = \mathbf{m}_k^{(t)} + \alpha^{(t)} h_{k,c}^{(t)} \left(\mathbf{y}^{(t)} - \mathbf{m}_i^{(t)} \right), \tag{4.4}$$

where t is time, $\mathbf{y}^{(t)} \in \mathbf{Y}$ is randomly selected at time t, and $0 \leq \alpha^{(t)} \leq 1$ denotes the learning rate.

4.3.2 The batch-SOM algorithm and its magnification factors

The original version of the SOM algorithm makes a separate update of the model parameters for each data point, taken one at a time, whereas its batch version, called batch SOM, makes the update on the basis of all data points.

Each data point is assigned to the region of the map to which is closest, according to the neighbourhood function $h(\cdot, \cdot)$. In this way the model defines a set of clusters in the data space.

The update of the weight vectors now follows the rule:

$$\mathbf{m}_k^{(t+1)} = \sum_{j=1}^{N} \frac{h^{(t)}(\mathbf{x}_k, \mathbf{x}_{k_j})}{\sum_{j'=1}^{N} h^{(t)}(\mathbf{x}_k, \mathbf{x}_{k_{j'}})} \mathbf{y}_j, \tag{4.5}$$

where $\mathbf{x}_{k_j}$ is the node corresponding to the BMU for $\mathbf{y}_j$. To improve the method, the data set is partitioned in each training step according to the m Voronoi regions V_j of weight vectors $\mathbf{m}_j$, each one containing n_{V_j} samples.

This update equation can be rewritten in a kernel regression form [Mulier and Cherkassky, 1995], for a given iteration, as:

$$\mathbf{m}_k = \sum_{k'} \left(F(\mathbf{x}_k, \mathbf{x}_{k'}) \bar{\mathbf{y}}_{k'} \right), \tag{4.6}$$

where $\bar{\mathbf{x}}_{k'} = \frac{1}{n_{V_k'}} \sum_{j \in V_k'} \mathbf{x}_j$ is the mean of the group $V_{k'}$ of $n_{V_k'}$ data points assigned to a given node k', and

$$F(\mathbf{x}, \mathbf{x}_k) = \frac{N_k h(\mathbf{x}, \mathbf{x}_k)}{\sum_{k'} N_{k'} h(\mathbf{x}, \mathbf{x}_{k'})}. \tag{4.7}$$

Magnification factors:

As stated in [Svensén, 1998] and [Bishop et al., 1997a], it is possible to explicitly calculate the magnification factor (MF) for the batch-SOM algorithm. We only need to compute the Jacobian of the mapping transformation in Eq. 4.6, whose columns

are given by

$$\mathbf{J}_{:,i} = \frac{\partial \mathbf{m}}{\partial x^{(i)}} = \sum_k \left(\frac{\partial F(\mathbf{x}, \mathbf{x}_k)}{\partial x^{(i)}} \bar{\mathbf{y}}_k \right) = \sum_k \frac{x^{(i)} - x_k^{(i)}}{\sigma^2} \left(F(\mathbf{x}, \mathbf{x}_k)^2 - F(\mathbf{x}, \mathbf{x}_k) \right) \bar{\mathbf{y}}_k. \tag{4.8}$$

The MF follows from Eq. 3.8

$$MF = \sqrt{det(\mathbf{J}^\top \mathbf{J})} \tag{4.9}$$

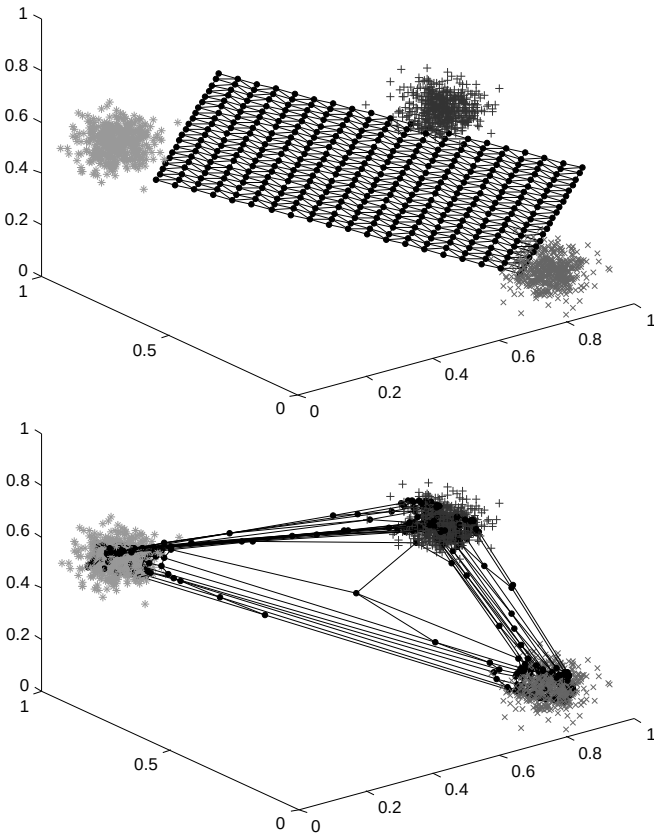


Fig. 4.3 Direct visualization of the artificial 3-D data, where each of the three clusters is represented in a different color. The display includes the overlaid embedded regular grid of batch-SOM reference vectors represented as a connected network. On the top: the regular grid is initialized as a linear projection of the data in 2-D. On the bottom: the grid is trained to fit the data with the batch-SOM algorithm. Note that this representation is only possible for 3-D observed data.

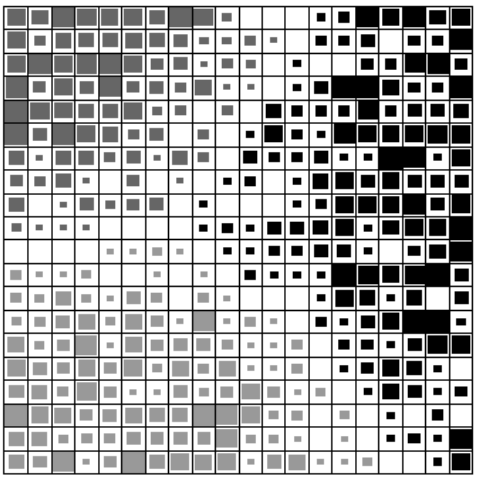


Fig. 4.4 Predefined square grid of the batch-SOM in the 2-D visualization space, where each cell corresponds to one map unit. Different colors are used to represent each of the three artificial data clusters. The cell area is proportional to the number of prototypes assigned to that cell.

4.3.3 Cartogram visualization for batch-SOM

We illustrate here the Cartogram representation of the MF for the batch-SOM model described in the previous section [Tosi and Vellido, 2012].

A first experiment involves the analysis of artificial data. A total of 1,500 3-D points are randomly drawn from 3 spherical Gaussian distributions (500 points each), all with unit variance, and with centres sitting at the vertices of a triangle. We choose 3-D data in order to explicitly allow the direct visualization of the reference vectors in the observed data space, to better understand the effects of the non-linear dimensionality reduction operated by SOM.

The batch-SOM algorithm is implemented in Matlab®, using the SOM-toolbox[2]. Data are preprocessed with both normalisation and scaling. We use a 20×20 rectangular regular grid for latent space and a Gaussian neighbourhood function. Figs.4.3 and 4.4 include, in turn, the direct visualization of the 3-D data together with the overlaid embedded regular grid of batch SOM reference vectors (which would not be available for data of higher dimensionality), and the standard batch-SOM map visualization in the form of a regular square grid of clusters.

The visualization space map reflects a neat but narrow separation between the

[2] www.cis.hut.fi/somtoolbox

three clusters which, in fact, are far from each other, as evidenced by the direct data visualization.

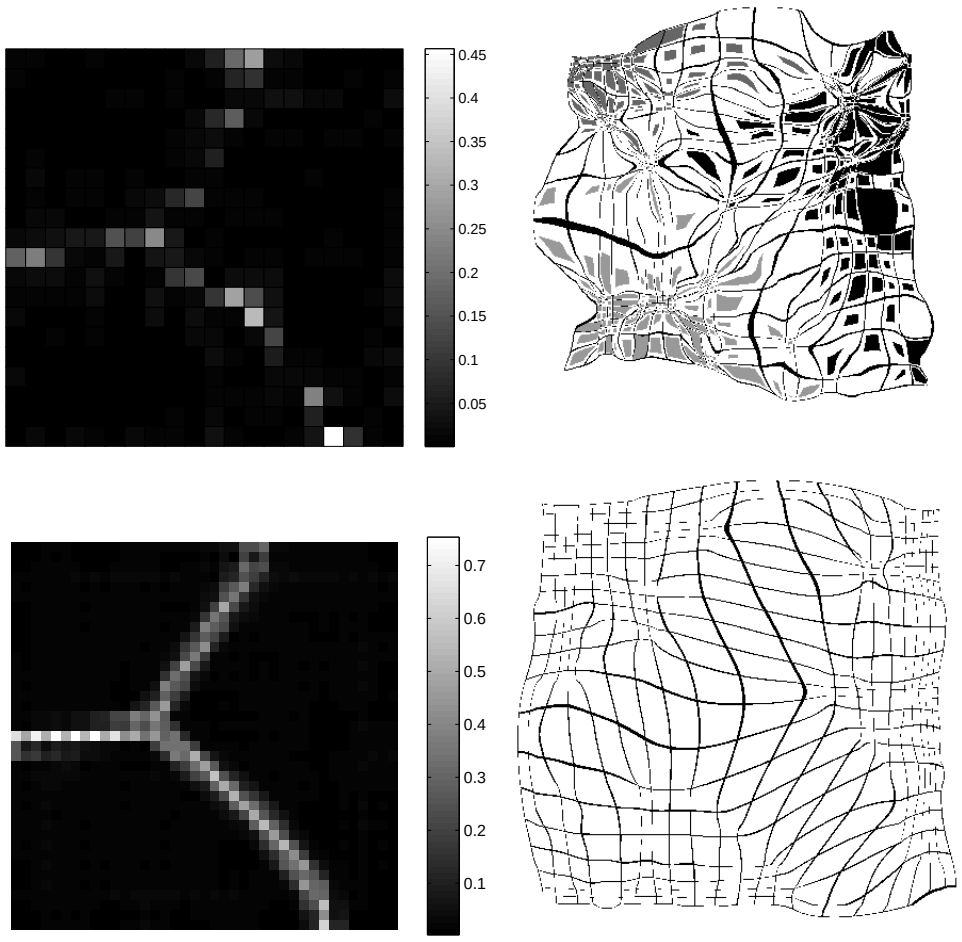


Fig. 4.5 Top row: left) Map of MF values together with a colorbar on the right-hand side of the map; right) corresponding Cartogram. Bottom row: left) U-matrix map; right) corresponding Cartogram.

We compute the MF value for each node $\mathbf{x}_k$ using Eq. 4.8. In the experiments, the batch-SOM maps are transformed into a Cartogram by using the rectangular grid, defined by the nodes $\mathbf{x}_k$, as map internal boundaries (effectively, defining a centroidal Voronoi tesselation [Du et al., 1999]). As described in section § 4.2, we assume that the level of distortion $d^{external}$ in the space beyond this rectangle is uniform and equal to the mean distortion over the complete map.

A similar procedure is applied to compute the Cartogram using the unified distance matrix as the underling distortion measure. The U-matrix [Ultsch, 2003] allows the visualization of the pairwise distances between corresponding points in the original data space on the low-dimensional visualization space of the SOM topographic

grid.

Results are displayed in Fig. 4.5. The overlaid grid of reference vectors seen in Fig. 4.3 explains the fact that many reference vectors are squashed in data-dense regions whereas only a few are stretched to cover the empty space in-between. This varying distortion is nicely reflected by both the MF and the U-matrix, on the left column of Fig. 4.5.

The batch-SOM map and the distortion measures finally come together in the Cartograms (MF: top-right, and U-matrix: bottom-right of Fig. 4.5. The empty spaces between clusters are now fairly stretched, providing a clear view of the separation.

Interestingly, part of the data reside in stretched areas: These are the ones further from the cluster centres. This effect should warn us against a too straightforward interpretation of high-distortion areas as completely empty spaces.

The visualization of the MF on the batch-SOM map may inform us of the existence of data clusters and the sparsely populated spaces that separate them, as they undergo different levels of distortion: low in dense areas, while high in empty ones. In this task, it is a principled alternative to the widely used U-Matrix [Ultsch, 2003]. This direct visualization is not always intuitive. Instead, the Cartogram-based representation of the batch-SOM map retains its simplicity while visually factoring out the non-linear distortion as measured by the MF.

4.4 Cartogram representations for GTM

The same idea proposed above for batch-SOM algorithm can be applied to the Generative Topographic Mapping (GTM), cfr section 2.2. In the following section, we propose here to explore some artificial data sets with simple statistical properties, in order to assess the properties of the method in a controlled setting.

We will work with 3-D data divided into three neatly defined clusters, to which atypical cases or outliers will be added. The dimensionality of data is chosen, again, to allow the direct visualization of prototypes embedded in the observed data space. Following the line of the experiment described in § 4.3, we explicitly calculated the MF for the t-GTM variant of the standard GTM and we introduce a Cartogram visualization of the latent space.

The standard GTM algorithm was implemented in Matlab®, using the *drtoolbox*[3].

[3]http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

4.4.1 Robust topographic mapping and its magnification factors

One constraint of the basic GTM model is due to the fact that the centres of the mixture components do not move independently from each other, as they are limited by definition to lie in a low-dimensional embedded manifold. The basic GTM model presented in § 2.2.1, has some obvious limitations when dealing with atypical data or *outliers*, due to the narrowness of the tails of the Gaussian distributions. The point is that the presence of outliers is likely to bias the estimation of parameters $\mathbf{W}$ and β, so other more robust formulations of GTM has been proposed using a mixture of Student's *t*-distributions (the t-GTM model). This model, when applied to multivariate data clustering and visualization, provides a more accurate imputation of missing values and it is robust when dealing with outliers [Vellido et al., 2006; Vellido, 2006b].

To introduce the multivariate t-GTM model, we assume that the basis functions ϕ_m in the GTM mapping given by Eq. 2.5 are replaced by *Student t*-distributions.

The conditional distribution of the data given the latent variables and the adaptive parameters is:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \beta\nu) = \frac{\Gamma\left(\frac{\nu}{2} + \frac{p}{2}\right)\beta^{p/2}}{\Gamma\left(\frac{\nu}{2}\right)(\nu\pi)^{p/2}} \left(1 + \frac{\beta}{\nu}\sum_{j=1}^{p}\left(\mathbf{y}_j - \sum_{m=i}^{M}\mathbf{w}_j^{\top}\phi_m(\mathbf{x})\right)^2\right)^{-(\nu+p)/2} \tag{4.10}$$

where Γ is the gamma function

$$\Gamma(t) = \int_0^{\infty} x^{t-1}e^{-x}\,\mathrm{d}x \tag{4.11}$$

and ν is an adaptive parameter such that the multivariate *t*-distributions converges to a multivariate normal one when $\nu \to \infty$.

We can now integrate the latent variables out. After this, we obtain new expressions of the log-likelihood and again the model can be fitted to data using the EM algorithm to obtain, along with other results, the responsibilities r_{kn}. For more details, see Vellido [2006a,b].

To update the parameters $\mathbf{W}$ and β, we apply the maximization step of the algorithm by maximizing the expected log-likelihood, but a similar update cannot be applied to the parameter ν. To update ν, we have to consider alternative approaches such as, for instance, running experiments for a range of its possible values, selecting

the best choice out them.

Svensèn [1998] gives an interpretation of the update expression of β as the off-manifold variance of the model being updated to the average weighted distance between original data and prototypes. This update formula, together with the one of the responsibility, implies the existence of a further weighting term for the *t*-GTM, which will be small for data outliers, as stated in [Peel and Mclachlan, 2000]. As a result, the influence of outliers on the model parameters will be effectively minimized [Vellido et al., 2006].

Magnification factors

The t-GTM generates, as its standard counterpart, a varying local distortion that can make exploratory data visualization difficult. This distortion can again be quantified over the latent space continuum with MF, as explained in § 3.3.

The analytical quantification of the MF can be expressed in terms of the derivatives of the basis functions ϕ_m as

$$MF = \sqrt{\det\left((\mathbf{W}\mathbf{\Psi})^\top \mathbf{W}\mathbf{\Psi}\right)}, \tag{4.12}$$

where $J = \mathbf{W}\mathbf{\Psi}$ is the Jacobian of the mapping transformation and $\mathbf{\Psi} \in \mathbb{R}^{M\times q}$ has elements ψ_{mi} defined as

$$\frac{\partial \psi_m(\mathbf{x})}{\partial x^{(i)}} = \frac{\Gamma(\frac{\nu+p}{2})(-\nu-p)\beta^{\frac{p+2}{2}}}{\Gamma(\frac{\nu}{2})\pi^{p/2}\nu^{\frac{p+2}{2}}}\left(x^{(i)} - \mu_m^{(i)}\right)\left(1 + \frac{\beta}{\nu}\|\mathbf{x} - \boldsymbol{\mu}_m\|^2\right)^{-\frac{\nu+p-2}{2}} \tag{4.13}$$

where μ_m, $m = 1, \ldots, M$ are the centres of the Student t-distributions.

Remark

Notice that the computation of the MF for the basic GTM can be done in the same way according to Eq. 4.12. The only change that need to be done is to replace the derivatives of the the Student-t functions with the derivatives of the Gaussian radial basis functions.

4.4.2 Cartogram visualization for t-GTM

The following experiments compare the effect of outliers on the Cartogram representations of the MF for the standard GTM and for t-GTM. An artificial data set of

3-D points is used to make possible the direct visualization of the data vectors $\mathbf{y}_k$ in the observed data space. A total of 1,500 3-D points are randomly drawn from 3 spherical Gaussian distributions (500 points each), all with unit variance and with centres set at the vertices of an equilateral triangle. Two different subsets of outliers are added to this data set:

- *A-type*): three outliers located on the normal to the imaginary plane defined by the cluster triangle that passes through its barycenter;
- *B-type*): three outliers located on the normal to one vertex of the imaginary triangle.

We choose a 15×15 regular grid of 2-D latent points. Both GTM and t-GTM are trained with the same initialization. The MF is calculated for both methods and Cartograms are generated using these values. In all cartograms, we assume assumed a homogeneous level of distortion in the space beyond the grid: this value is chosen to be equal to the mean distortion over the complete map $1/K \sum_{k=1}^{K} MF_{\mathbf{x}_k}$, as described in section § 4.2. Likewise, we assume that the level of distortion within each of the squares associated to $\mathbf{x}_k$ is itself uniform.

The first experiment, displayed in Fig. 4.6, corresponds to the inclusion of *A-type* outliers, while the second, displayed in Fig. 4.7, corresponds to the inclusion of *B-type* outliers.

Despite the fact that most GTM prototypes concentrate in the three clusters, it is clear from the image in Fig. 4.6 (top row, left) that, in the case of standard GTM, the *A-type* outliers force the manifold towards them in an undue manner. This causes a distortion that is more controlled by the outliers than by the empty space between clusters. Even though, the Cartogram visualizations generated by GTM and t-GTM are rather similar. The reason for this is the artificial symmetry of the outliers location.

The maps in Fig. 4.7, corresponding to the second experiment with added $B - type$ outliers, tell a very different story. Now, the symmetry is lost and the MF of the standard GTM reflects the fact that the model stretches one of the sides of the manifold in its attempt to cover the outliers (top row, left). As a result, an artefactual high distortion appears in the top-righ corner of the MF representation map (where outliers are seen to be mapped) and biases the Cartogram representation. The t-GTM, instead, ignores the outliers and respects the symmetry of the representation while restricting the manifold to the imaginary triangle defined by the three clusters. This is clearly reflected in the corresponding Cartogram.

Notice that, in both experiments, the extra MF distortion introduced by the outliers makes the data representation of the data of all clusters far more compact for GTM than for t-GTM. In any case, this simple preliminary experiments illustrate how modelling methods that behave robustly in the presence of outliers are more likely to produce more faithful representations of the non-linear mapping distortion and, as a result, more faithful data visualizations.

4.5 Discussion

In this chapter we have presented a novel technique that allows to introduce into the visualization space a quantitative representation of the non-linear mapping distortion due to the chosen dimensionality reduction model. One advantage of this technique is its *portability*, since a Cartogram visualization is possible for any non-linear dimensionality reduction method in which a distortion measure can be defined.

The experiments presented here are meant to be exploratory, aiming to asses the validity of the proposed model. Further investigation can be done by using more complex datasets, with different characteristics and different statistical properties. One example of this is presented in the recent work of Vellido et al. [2013], where the results over diverse artificial datasets provide some guidelines for the use of Cartograms in NLDR. It is shown that some underlying quantities, such as the dimensionality of the data features, affect the visualization space more than others, such as the density of data clusters or the size of the latent grid.

It is important to stress that the use of Cartogram visualization does not affect or modify the performance of the model during its training. It is just an *a posteriori* method for the visual display of the results. The Cartograms provide added value to the visualization of the low-dimensional space by enabling a deeper analysis of the model capabilities. One example of this has been given in section 4.4, where we have shown that the Cartogram representation reflects the model capability of behaving robustly in the presence of atypical data.

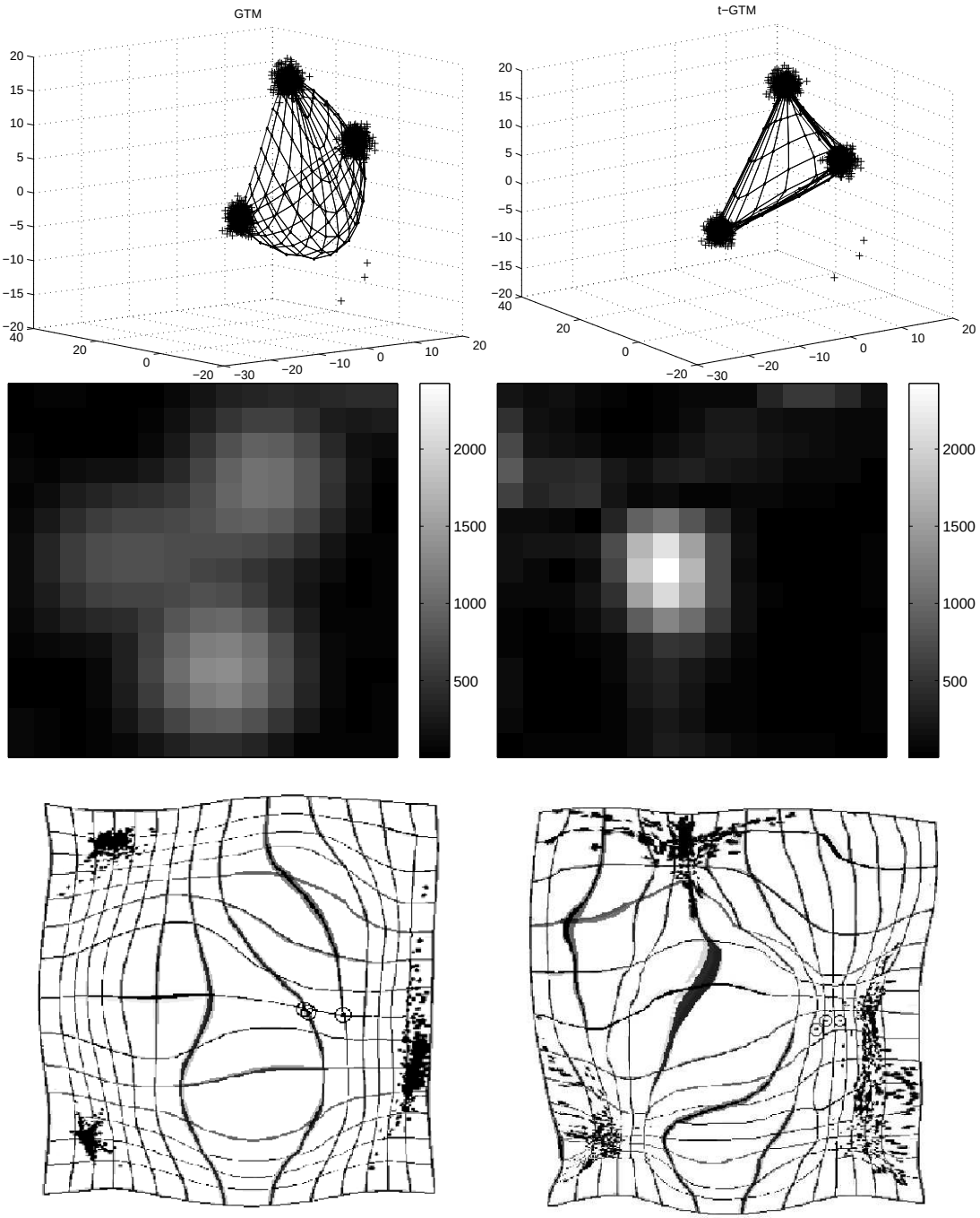


Fig. 4.6 Top row) Representation of the original data clusters (1,500 points, 500 in each cluster, plus $A - type$ outliers, all represented with crosses) with standard GTM (left) and t-GTM (right) The generated manifold is superimposed; it is represented as a grid whose knots are the model prototypes $\mathbf{y}_k$; central row) Maps of MF values together with a colorbar for interpretation on the right-hand side of the maps; bottom row) Corresponding Cartograms, based on the MF, to which the mean projections of the data are superimposed. The mapping locations of outliers are highlighted with circles.

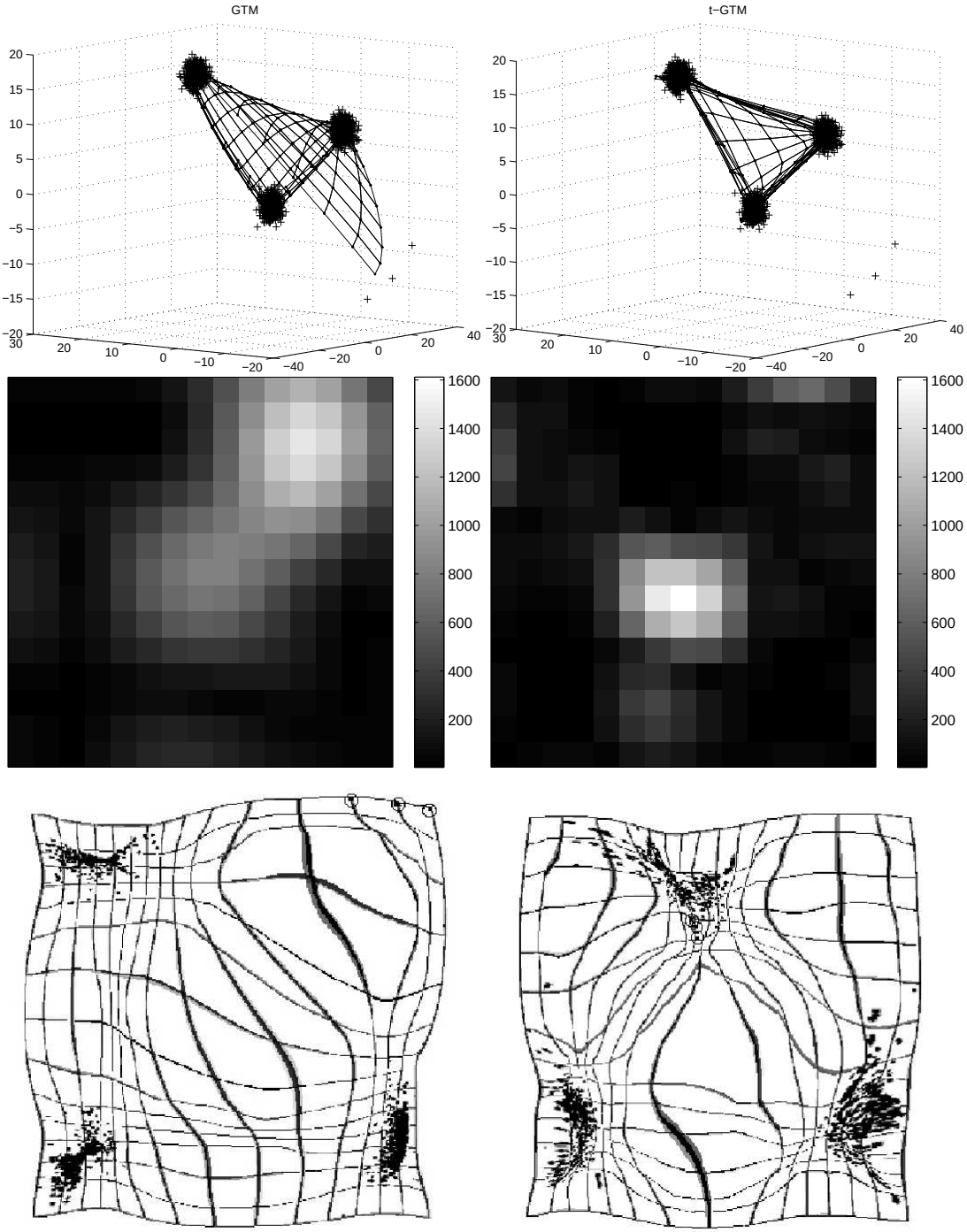


Fig. 4.7 Representation of data, manifold grid, MF maps and Cartograms for the second experiment with $B-type$ outliers as in Fig. 4.6. Notice the difference in the mapping locations of outliers (again highlighted with circles) as compared to Fig. 4.6. In this case, GTM maps the outliers in a high-distortion area that is generated by the own outliers and not by the cluster data points (note that this high distortion appears because of just three outlier points), whereas the t-GTM maps them correctly to the closest cluster, without any artefactual distortion.

Chapter 5

Increasing Interpretability of MTS Modelling through Visualization Using Manifold Learning

Time-dependent natural phenomena and artificial processes can often be quantitatively expressed as multivariate time series (MTS). As in any other process of knowledge extraction from data, the analyst can benefit from the exploration of the characteristics of MTS through data visualization. This visualization often becomes difficult to interpret when MTS are modelled using non-linear techniques. Despite their flexibility, non-linear models can be rendered useless if such interpretability is lacking.

The methods described in previous chapters have mostly focused on static i.i.d. data. In this chapter, we model MTS using VB-GTM-TT, a variational Bayesian variant of a constrained hidden Markov model (HMM) of the manifold learning family defined for MTS visualization. We aim to increase its interpretability by taking advantage of two results of the probabilistic definition of the model: the explicit estimation of probabilities of transition between states described in the visualization space and the quantification of the non-linear mapping distortion.

5.1 Exploring MTS

Most applied analysis of multivariate time series involves, in one way or another, problems with specific targets such as prediction, forecasting, or anomaly detection. A less explored avenue of research is the exploratory analysis of multivariate time series using machine learning and computational intelligence methods [Fu, 2011].

Data exploration may be a key stage in knowledge extraction from multivariate time series using complex non-linear methods, as it opens the door to their interpretability [Vellido et al., 2012].

As in any other processes of knowledge extraction from data, the analyst could benefit from the exploration of the characteristics of temporal data consisting of a high number of individual series through their visualization [Vellido et al., 2011]. The direct visualization of such high-dimensional data, though, can easily be beyond the interpretation capabilities of human experts. Therefore, the exploration of temporal data can be assisted by dimensionality reduction methods. In particular, the visualization of multivariate time series using non-linear dimensionality reduction (NLDR) methods [Lee and Verleysen, 2007] can provide the expert with inductive reasoning tools as a means to hypothesis generation. Visualization can thus facilitate interpretation, which is paramount given that NLDR methods can be rendered useless in practice if interpretability is lacking.

In this chapter, we merge two strands of previous research on data visualization. The first one involves the visualization of multivariate time series using statistical machine learning and NLDR methods [Bishop et al., 1997b]. The second tackles one of the main interpretability bottlenecks of NLDR techniques: the difficulty of expressing the non-linear mapping distortion they introduce in the data visualization space in an intuitive manner. Specifically, we attempt to increase the interpretability of the variational Bayesian generative topographic mapping (VB-GTM-TT), a variational Bayesian variant of a constrained hidden Markov model (HMM) [Rabiner, 1989] of the manifold learning family, defined for the visualisation of multivariate time series [Olier and Vellido, 2008a]. For this, we use two results of the probabilistic definition of the model: the explicit estimation of probabilities of transition between states described in the visualization space and the quantification of the distortion introduced by the non-linear mapping of the multivariate time series in the form of magnification factors (MFs).

Note that our analysis does not address the assessment of the quality of the mapping as such. In fact, the proposed visualization strategies are meant to be independent from it. Although VB-GTM-TT is used here for illustration (as a method that, even if prone to limitations such as local minima, has been shown to perform robustly in the presence of noise), the proposed approach could be extended to alternative dimensionality reduction models for multivariate time series, for which distortion and probability of state transition (or some approximations to them) are quantifiable.

5.2 Variational Bayesian GTM Through Time

The standard GTM model has been reformulated within a variational full Bayesian framework by Olier and Vellido [2008b], providing an extention to the analysis of MTS in Olier and Vellido [2008a]. The result is the VB-GTM-TT: a model that integrates regularization explicitly and provides adaptive optimization of most of the model parameters involved.

Assuming a sequence of hidden states $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1,\ldots,N}$ for every time step n and the observed MTS $\mathbf{Y} = \{\mathbf{y}_n,\}_{n=1,\ldots,N}$, the complete-data likelihood for VB-GTM-TT is given by:

The model parameters are $\boldsymbol{\Theta} = (\pi, \mathbf{A}, \mathbf{U}, \beta)$, specified by

$$\text{initial state probabilities}: \quad \pi = \{\pi_j\} : \pi_j = p\left(\mathbf{z}_1 = j\right) \tag{5.1}$$

$$\text{transition state probabilities}: \quad \mathbf{A} = \{a_{ij}\} : a_{ij} = p\left(\mathbf{z}_n = j | \mathbf{z}_{n-1} = i\right) \tag{5.2}$$

$$\text{emission probabilities}: \quad \{\mathbf{U}, \beta\} : p\left(\mathbf{y}_n | \mathbf{z}_n = j\right) = \left(\frac{\beta}{2\pi}\right)^{p/2} \exp\left(-\frac{\beta}{2}\left\|\mathbf{y}_n - \mathbf{u}_j\right\|^2\right) \tag{5.3}$$

$$\text{inverse variance}: \quad \beta \tag{5.4}$$

The emission probabilities are controlled by spherical Gaussian distributions with common inverse variance β and a matrix $\mathbf{U}$ of K centroids $\mathbf{u}_j$, $1 \leq j \leq K$. They can be considered as hidden variables and integrated out to describe the marginal likelihood as:

$$p\left(\mathbf{Z}, \mathbf{Y}\right) = \int p\left(\boldsymbol{\Theta}\right) p\left(\mathbf{Z}, \mathbf{Y} | \boldsymbol{\Theta}\right) d\boldsymbol{\Theta}, \quad \text{where } \boldsymbol{\Theta} = \left(\boldsymbol{\pi}, \mathbf{A}, \mathbf{U}, \beta\right). \tag{5.5}$$

VB-GTM-TT assumes its parameters to be independent, so that

$$p(\boldsymbol{\Theta}) = p(\boldsymbol{\pi}) p(\mathbf{A}) p(\mathbf{U}) p(\beta),$$

where the set of prior distributions $p\left(\boldsymbol{\Theta}\right)$ are defined as:

$$
\begin{aligned}
p(\pi) &= \mathtt{Dir}\left(\{\pi_1, \ldots, \pi_K\}\,|\nu\right) \\
p(\mathbf{A}) &= \prod_{j=1}^{K} \mathtt{Dir}\left(\{a_{j1}, \ldots, a_{jK}\}\,|\lambda\right) \\
p(\mathbf{U}) &= \left[(2\pi)^K\,|\mathbf{K}|\right]^{-p/2} \prod_{i=1}^{p} \exp\left(-\frac{1}{2}\mathbf{u}^{(i)T}\mathbf{K}^{-1}\mathbf{u}^{(i)}\right) \\
p(\beta) &= \Gamma\left(\beta|d_\beta, s_\beta\right).
\end{aligned}
$$

Here, $\mathtt{Dir}(\cdot)$ represents the Dirichlet distribution and $\Gamma(\cdot)$ is the Gamma distribution. The vector ν, the matrix λ and the scalars d_β and s_β correspond to the hyperparameters of the model which are fixed *a priori*. The prior over the parameter $\mathbf{U}$ defines the mapping from the hidden states to the data space as a Gaussian Process (GP), where $\mathbf{u}_{(d)}$ is each of the row vectors (centroids) of the matrix $\mathbf{U}$ and $\mathbf{K}$ is a matrix where each element is defined by the covariance function as:

$$
\mathbf{K}_{i,j} = c\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\alpha^2}\right), \quad i, j = 1 \ldots K.
$$

The α parameter controls the flexibility of the mapping from the latent space to the data space. The vector $\mathbf{x}_j,\ j = 1 \ldots K$ corresponds to the state j in a latent space of usually lower dimension than that of the data space (for MTS visualization purposes). Thus, a topography over the states is defined by the GP as in the standard GTM. The VB-GTM-TT is optimized using variational approximation techniques. A more detailed description of the VB-GTM-TT and its formulation is provided in [Olier and Vellido, 2008a; Olier, 2008].

5.2.1 The VB-GTM-TT model and its magnification factors

As stated in § 3.3, the magnification factor (MF) can be explicitly computed for the batch-SOM (c.f. Eq. 4.8) and GTM (c.f. Eq. 4.12). In this section, we introduce here the calculation of the MF for the VB-GTM-TT model. For this, we first consider the jointly Gaussian random variables

$$
\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_{(*,\cdot)} \\ \mathbf{K}_{(\cdot,*)} & \mathbf{K}_{(*,*)} \end{bmatrix}\right), \tag{5.6}
$$

where $\mathbf{y}_*$ is a test point and $C_{(\cdot,\cdot)}$ is the covariance matrix defined according to (5.6). Due to the properties of Gaussian distributions, we can explicitly write the posterior

probability as follows:

$$\mathbf{y}_*|\,\mathbf{x}_*, \mathbf{\Theta} \sim \mathcal{N}\left(\mathbf{y}_* \mid \mathbf{K}_{(*,\cdot)}\mathbf{K}^{-1}\mathbf{U},\ \ \mathbf{K}_{(*,*)} - \mathbf{K}_{(*,\cdot)}\mathbf{K}^{-1}\mathbf{K}_{(\cdot,*)}\right). \tag{5.7}$$

The Jacobian J of this mapping can be obtained by computing the derivatives of $\langle(\mathbf{y}_*|\,\mathbf{x}_*, \mathbf{\Theta})\rangle$ with respect to x, using:

$$\frac{\partial k_{(*,j)}}{\partial \mathbf{x}_*^l} = \frac{1}{\alpha^2}(x_*^l - x_j^l)\exp\left(-\frac{\|\mathbf{x}_* - \mathbf{x}_j\|^2}{2\alpha^2}\right), \quad l = 1\ldots q, j = 1\ldots K,$$

being q the dimension of the latent space. As a result, the MF is calculated as:

$$MF = det^{-\frac{1}{2}}\left(J^T J\right) \tag{5.8}$$

The MF does not only provide us with a quantification of the local mapping distortion that separates areas of the visual map which have undergone much compression or stretching from those which have not; it also tells us about data sparsity: the model distorts the most in areas which are mostly empty of data and the least in densely populated areas. For this reason, the MF has been used as an indicator of the existence of data clusters and the boundaries between those clusters [Tosi and Vellido, 2013]. For MTS, we would expect the time series to flow over time through areas of low MF mostly when the MTS evolve slowly, whereas fast transitions between MTS regimes might require crossing areas of higher distortion.

5.2.2 Cumulative state transition probabilities

Another metric that might help improving the interpretability of the mapping is the likelihood for a state to be transited by any of the potential trajectories through states. Again, this can explicitly be quantified, for each state j defined by VB-GTM-TT, as the estimated cumulative state transition probability ($CSTP$) defined as the sum of the probabilities of transition from all states to it

$$\mathcal{CSTP}_j = \sum_{i=1}^{K} a_{ij}, \tag{5.9}$$

where the transition state probabilities are defined according to Eq. (5.2).

We would expect the MTS trajectory to pass through areas of high $CSTP$, because these should be areas of highly likely transition. As such, the $CSTP$ plays the opposite role to MF, because the areas of large manifold stretching (high MF) should

mostly be areas that the MTS is unlikely to cross (low $CSTP$).

5.3 Experiments and discussion

5.3.1 Materials and experimental setup

We illustrate the proposed MTS visualization using two different datasets. The first is an *artificial* 3-variate time series, with $1,000$ time points. The second set is the *Shuttle-data* from Space Shuttle mission STS-57[1]: a time series consisting of 1,000 points described by 6 features. This data set has previously been used for cluster detection [Lin et al., 2004].

5.3.2 MTS Visualization

The considered MTS are particularly suitable for the illustration of the proposed visualization techniques due to the nature of their regimes and transitions periods. The *artificial* dataset, displayed in Fig. 5.1 (top-row, left), is characterized by two intervals with regular regimes, divided by a sudden transition at point 700. The VB-GTM-TT model was trained over a 8×8, $2-D$ grid of hidden states and each of the MTS points was mapped by VB-GTM-TT to a particular state in the grid.

The result of this mapping assignment is shown in Fig. 5.1 (top-row, right). Before point 700, the periodicity of the data is well-captured by the roughly circular structure of populated states. The sudden transition to a higher-amplitude periodic interval is also neatly visualized.

On the other hand, *Shuttle_Data* presents four periods of little variability A-C-D-E and one period of high (quasi-periodic) variability B, which are separated by sudden transitions, as evidenced by their display in Fig. 5.2 (top row, left).

The VB-GTM-TT model was trained over a 13×13 grid of hidden states and each of the MTS points was mapped by VB-GTM-TT to a particular state in the grid, as shown in Fig. 5.2 (top row, right). There is a clear interpretation for this state membership mapping, as the *Shuttle-data* trajectory is confined to a limited number of its states (a common characteristic of VB-GTM-TT mappings, in which over-complexity is penalized). Only a few of them are relatively big: these are mostly stationary states with little MTS change in intervals C, D and E. The quasi-periodic interval B evolves slowly through a cloud of states on the top-left and center of the map.

[1]Which can be downloaded under request from www.cs.ucr.edu/∼eamonn.

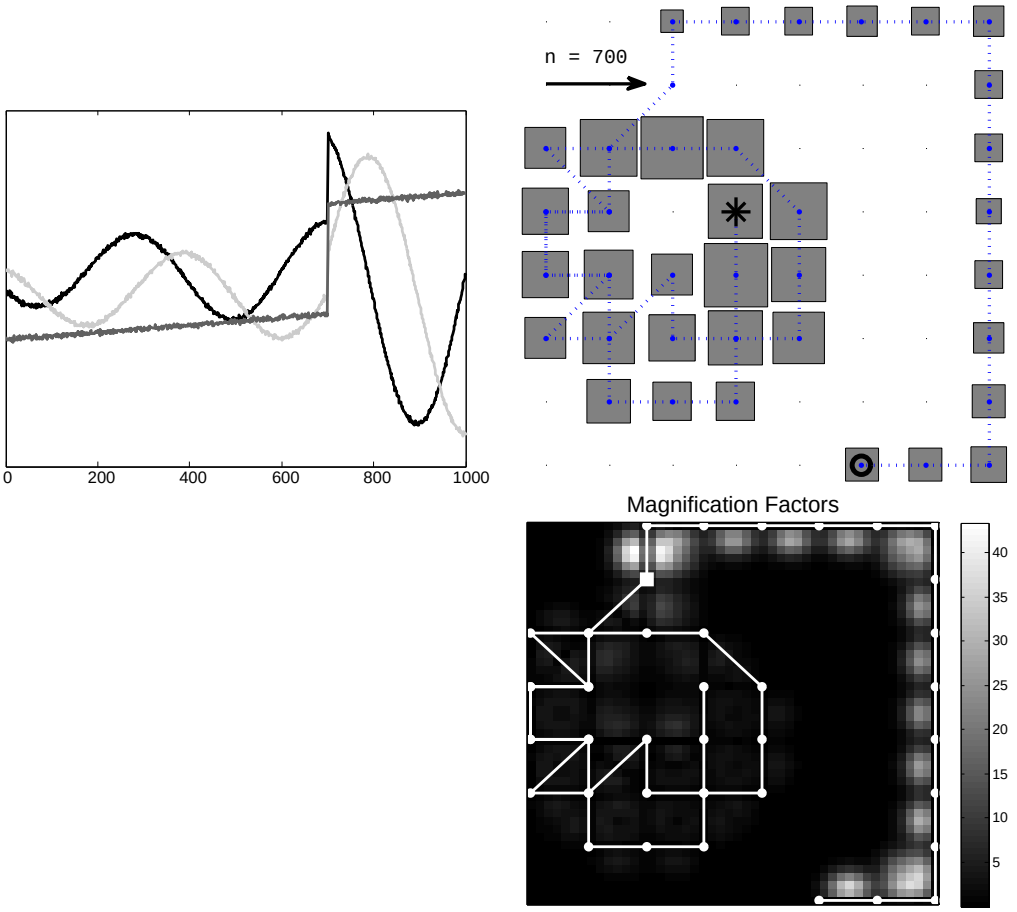


Fig. 5.1 Top row, left: Artificial dataset: a 3-variate time series, characterized by a sudden transition at $n = 700$. Top row, right: State-membership map of VB-GTM-TT, with a 8×8 grid of hidden states represented as squares, whose relative size is proportional to the time data points assigned to them; the starting point of the MTS is represented as a star and the ending point is represented as a circle. The sudden transition point is signalized by an arrow. Bottom row, right: MF gray-shade color map, represented in the VB-GTM-TT latent space visualization grid. The trajectory of the MTS over the map is displayed as a white solid line.

The MFs were computed for *artificial* and *Shuttle-data* and represented in Figs. 5.1 and 5.2 (bottom, right) through color maps over the grid of hidden states. For both datasets, it might seem at first sight that the MTS cross through areas of high MF (high distortion), a behaviour that would refute the hypothesis that the densely data populated areas correspond to low mapping distortion. In fact, this is not the case: the MTS mostly flows over time through *channels* of low distortion surrounded by borders of high distortion. These borders seem to act as barriers that compel the MTS to follow a given trajectory. In fact, these barriers are only breached (with the MTS moving briskly towards higher MF) in sudden transitions between regimes. These can clearly be seen for *Shuttle-data* if we plot the value of MF over time, as in Fig. 5.2

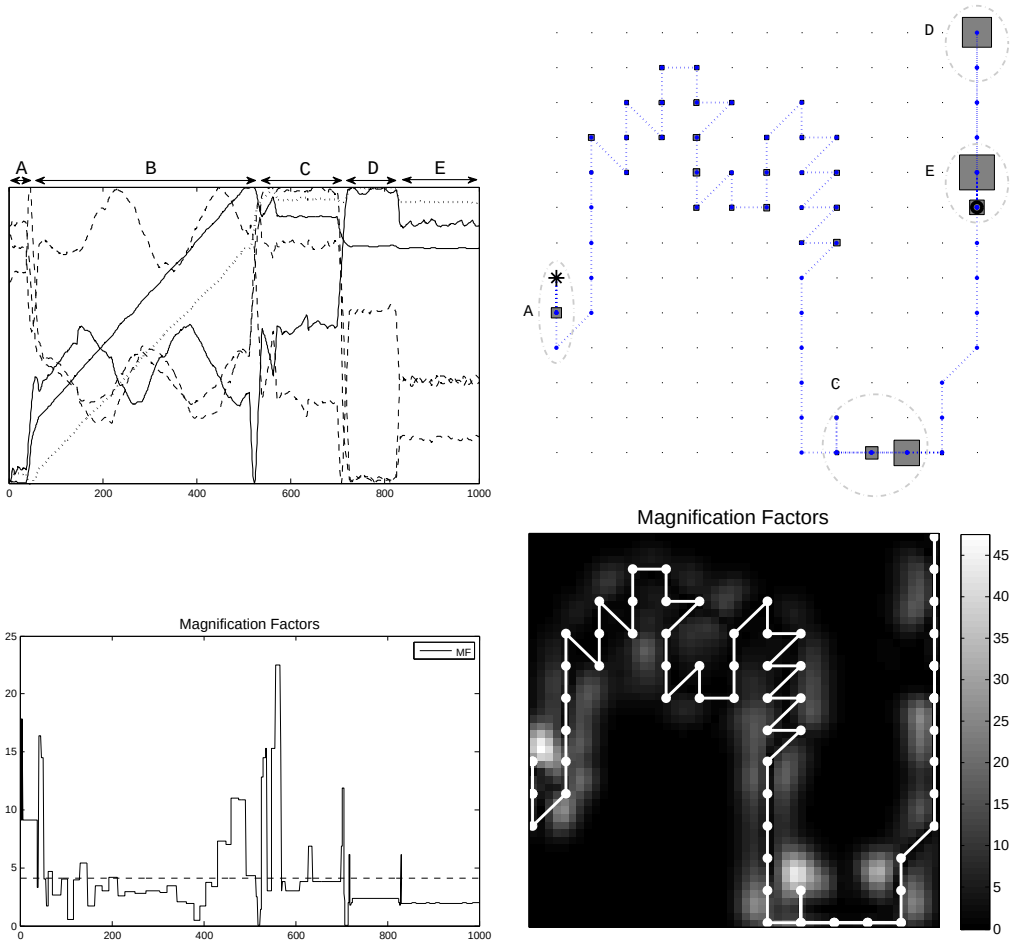


Fig. 5.2 Top left: Plot of *Shuttle-data*; the five intervals or regimes separated by sudden transitions are identified as A, B, C, D and E. Top right: State-membership map generated by VB-GTM-TT, with a 13×13 grid of hidden states represented as squares; the relative size of these squares is proportional to the time data points assigned to them; the starting point of the MTS is represented as a star, the ending point as a circle. Bottom Left: The Magnification Factors as a function of time, including the mean MF over all states (represented as a dashed line); narrow peaks of distortion are detected precisely in the areas of sudden transitions. Bottom Right: MF gray-shade color map, represented in the VB-GTM-TT latent space visualization grid; white areas correspond to high distortion.

(bottom row, left): MF narrow spikes of varying magnitude (particularly strong in the transition from B to C) appear in the transitions between time intervals. These spikes take values well over the mean MF of the map. This result suggests that the evolution of the MF over time could directly be used to detect sudden regime transitions in MTS.

The $CSTP$ maps in Fig. 5.3 are very consistent with their MF counterparts, and complement them. Alternatively displayed as $3 - D$ maps over the grid of hidden states, they provide an intuitive illustration of the previously described behaviour.

Following a geographical representation visual metaphor, the MTS can be seen to flow across cumulative state transition probability *ridges*, where rapid transitions between regimes see the MTS moving through relatively lower-valued *depressions* in those *ridges*. An opposite graphical metaphor could be used for the MF distortion, with the MTS flowing through its *valleys*, that is, across areas of the map characterized by low MF values.

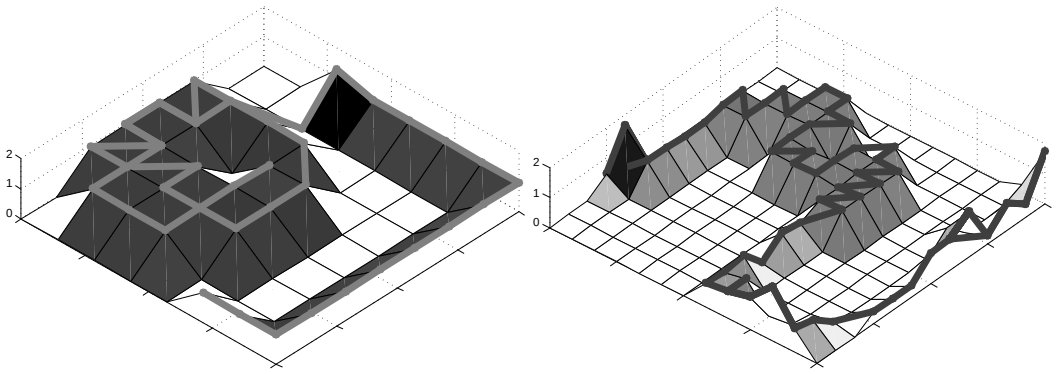


Fig. 5.3 A $3-D$ representation for the CSTP plots. The values in the vertical axis correspond to the $CSTP$ values over the latent space. Left: *artificial* data; right: *Shuttle-data*.

5.4 Discussion

Data visualization can be of great assistance in knowledge extraction processes. High dimensionality is always a barrier for visualization. In the case of MTS, this is compounded by their i.i.d. nature, because the search for patterns over time is often relevant in their study. Dimensionality reduction can make visualization operative for high-dimensional MTS. The use of non-linear dimensionality reduction methods to this purpose poses a challenge of model interpretability due to the existence of locally-varying distortion.

In this chapter, we have proposed to use MF and $CSTP$ to improve interpretability for VB-GTM-TT, a manifold learning NLDR method. The model mapping distortion has been explicitly quantified in the latent space continuum and the probabilistic nature of the method has allowed us to define a cumulative probability of state transition. The reported experiments have shown that both metrics can provide interesting insights that enhance the low-dimensional visualization of the MTS provided by the model.

This exploration approach is quite flexible and could be extended to other dimensionality reduction models for MTS analysis, provided their local distortion can be

quantified. Examples of this may include GP-LVM [Lawrence, 2005]), GP dynamical models (GPDM, [Wang et al., 2008; Damianou et al., 2011]) or temporal Laplacian eigenmaps [Lewandowski et al., 2010]. It could also be extended to alternative visual display methods, such as the Cartograms presented in chapter § 4, [García et al., 2013; Tosi and Vellido, 2013, 2012] and warped topographic maps [Gianniotis, 2013].

Chapter 6

Metrics for Probabilistic Geometries and Their Impact on Interpretability

In many practical applications of multivariate data analysis, probabilistic models are combined with heuristic algorithms that use computed distances and interpolants between observations; this is, for instance, the case of distance-based methods for classification and clustering, methods of data interpolation for the reconstruction of missing frames (in computer vision problems), or techniques for the definition of the optimal path from one data point to another (in robotics, or cartography), to name a few. These heuristic algorithms are often based on the assumption of a deterministic system, without taking into account the uncertainty deriving from the probabilistic framework. Therefore, if the data observations are noisy, should the underlying geometry (and hence distances and interpolants) not be considered noisy as well?

This chapter studies the role of Random Geometries, i.e. distributions of manifolds, in machine learning and, more generally, in data analysis. We develop methods for the estimation of distributions of Riemannian manifolds which support observations as well as algorithms for computing interpolants (geodesics) over distributions of manifolds.

The new geometrical insight given to the interpretation of probabilistic modelling may not only theoretically advance statistical machine learning, but also improve the usability and flexibility of existing models.

6.1 Metrics for Probabilistic LVMs

Manifold learning approaches attempt to learn the underlying support of the data (the manifold). Using the concepts of Riemannian geometry presented in section § 3.2, it is possible to derive the intrinsic geometrical properties of the model by explicitly computing its local metric tensor continuously over the input space. Once the metric has been derived, is then possible to compute geometrical quantities such as distances, angles, or the curvature of the space.

Given that manifold learning models can be of different nature, we have decided to restrict the developments in this chapter to smooth generative models. In such models, the local metric varies smoothly across the input space, in contrasts with prior approaches such as those reported in [Bregler and Omohundro, 1994; Tenenbaum, 1997; Tenenbaum et al., 2000], which use metrics that vary discretely across the space. Other relevant approaches related to Gaussian models can be found in [Lawrence, 2012].

In the following sections, the local metric tensor for generative latent variable models is first defined. We then illustrate the specific cases of GP-LVM and GTM, providing two algorithm to compute shortest paths (geodesics). The novelty of this approach is the probabilistic expression of the local metric, which opens to new streams of investigation in the field of probabilistic geometries for latent variable models.

6.2 The distribution of the natural metric

Given a noise model as in in Eq. (2.1), we assume the mapping f between the latent space and the observed space to be a differentiable function. This appears not to be a strict requirement, since some of the most used models fulfill it. Two examples are the standard GTM and the GP-LVM with the *exponentiated quadratic* kernel (EQ), since in both cases the differential of f reduces to the differential of an exponential.

Under this assumption of smoothness, the output of the mapping can be interpreted as a differential manifold (c.f. section § 3.2). It is then possible to explicitly compute the natural Riemannian metric of the given model as follows: let $\mathbf{J}$ be the Jacobian of the mapping f given as in Eq. (1.1), then the tensor

$$\mathbf{G} = \mathbf{J}^\top \mathbf{J}$$

defines a local inner product structure over the latent space according to Eq. 3.3.

Being the Gaussian distribution widely used in latent variable models, it is practical to analyse the particular case in which the conditional probability over the Jacobian follows a Gaussian distribution as well. The distribution over the Jacobian is inducing a distribution over the local metric tensor $\mathbf{G}$ in a natural way. In fact, assuming independent rows of $\mathbf{J}$, the distribution of the Jacobian is the product of p multivariate Gaussians

$$p(\mathbf{J} \mid \mathbf{X}, \mathbf{f}, \beta) = \prod_{j=1}^{p} \mathcal{N}(\mathbf{J}_{j,:} \mid \boldsymbol{\mu}_{J_{j,:}}, \boldsymbol{\Sigma}_J). \tag{6.1}$$

The resulting random variable follows a non-central Wishart distribution [Anderson, 1946]

$$\mathbf{G} = \mathcal{W}_q(p, \boldsymbol{\Sigma}_J, \mathbb{E}[\mathbf{J}^\top]\mathbb{E}\left[\mathbf{J}\right]), \tag{6.2}$$

where p represents the number of degrees of freedom; the quantity $\boldsymbol{\Sigma}_J^{-1}\mathbb{E}[\mathbf{J}^\top]\mathbb{E}\left[\mathbf{J}\right]$ is known as the non-centrality matrix and it is equal to zero in the central Wishart distribution. Intuitively, we can interpret the Wishart distribution as a multivariate generalisation of the Gamma distribution.

It is interesting to observe that the expectation of the metric tensor is given by the sum of two terms, a mean term and a covariance term

$$\mathbb{E}[\mathbf{J}^\top\mathbf{J}] = \underbrace{\mathbb{E}[\mathbf{J}^\top]\mathbb{E}[\mathbf{J}]}_{\text{mean term}} \quad + \quad \underbrace{p \cdot \boldsymbol{\Sigma}_J}_{\text{covariance term}} \quad , \tag{6.3}$$

whose role will be made more explicit at the end of the section.

Given the general formulation of the distribution of the Riemannian metric tensor in generative latent variable models, we can now provide the explicit expression for the models of our interest.

GP-LVM local metric

As described in section § 2.3, a Gaussian process (GP) can be used in dimensionality reduction to describe distributions over a mapping f

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

$$y_{i,j} = \mathbf{K}_{\mathbf{f}_i,\mathbf{f}}\mathbf{K}\mathbf{Y}_{:,j} + \epsilon_i,$$

leading to the formulation of the Gaussian process latent variable model (GP-LVM) Lawrence [2005].

It follows from Eq. 2.36 and the properties of the GPs that the distribution of the Jacobian of the GP-LVM mapping is the product of p independent Gaussian distributions (one for each dimension of the dataset) with mean $\boldsymbol{\mu}_{J(j,:)}$ and covariance $\boldsymbol{\Sigma}_J$. For every latent point $\mathbf{x}_*$, the Jacobian takes the following form:

$$p(\mathbf{J} \mid \mathbf{Y}, \mathbf{X}, \mathbf{x}_*) = \prod_{j=1}^{p} \mathcal{N}(\mathbf{J}_{j,:} \mid \boldsymbol{\mu}_{J_{j,:}}, \boldsymbol{\Sigma}_J) \tag{6.4}$$
$$= \prod_{j=1}^{p} \mathcal{N}(\partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{Y}_{:,j}, \partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^{\top} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}),$$

which (c.f. Eq. 6.2) gives a distribution over the metric tensor $\mathbf{G}$

$$\mathbf{G} = \mathcal{W}_q(p, \partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^{\top} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}, \mathbb{E}[\mathbf{J}^{\top}]\mathbb{E}[\mathbf{J}]). \tag{6.5}$$

From this distribution, the expected metric tensor can be computed as

$$\mathbb{E}[\mathbf{J}^{\top}\mathbf{J}] = \mathbb{E}[\mathbf{J}^{\top}]\mathbb{E}[\mathbf{J}] + p \underbrace{\partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^{\top} \tilde{\mathbf{K}}_{\mathbf{f},\mathbf{f}}^{-1} \partial \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}}_{\text{covariance term}}. \tag{6.6}$$

Magnification Factors

The metric tensor defines the local geometric properties of the GP-LVM model and it can be used as a tool for data exploration that helps increasing the model interpretability. One way to visualise the tensor metric is through the differential volume of the high dimensional parallelepiped spanned by GP-LVM; this, for a latent dimension $q = 2$ is known as the Magnification Factor (MF), see section § 3.3. The explicit formulation of the MF for GP-LVM is given by

$$\mathrm{MF} = \sqrt{\det\left(\mathbb{E}[\mathbf{J}^{\top}\mathbf{J}]\right)}. \tag{6.7}$$

An illustrative example of MF visualization, using the *Shuttle* dataset presented in section § 5.3.1, is displayed in Fig. 6.1. The colormap is computed over a fine regular grid defined over the latent space. An additional example has been displayed in the background section § 3.3, using the jogging motion of the CMU motion capture database described in section § 2.3.3.

Further examples are provided in the experimental results of this chapter.

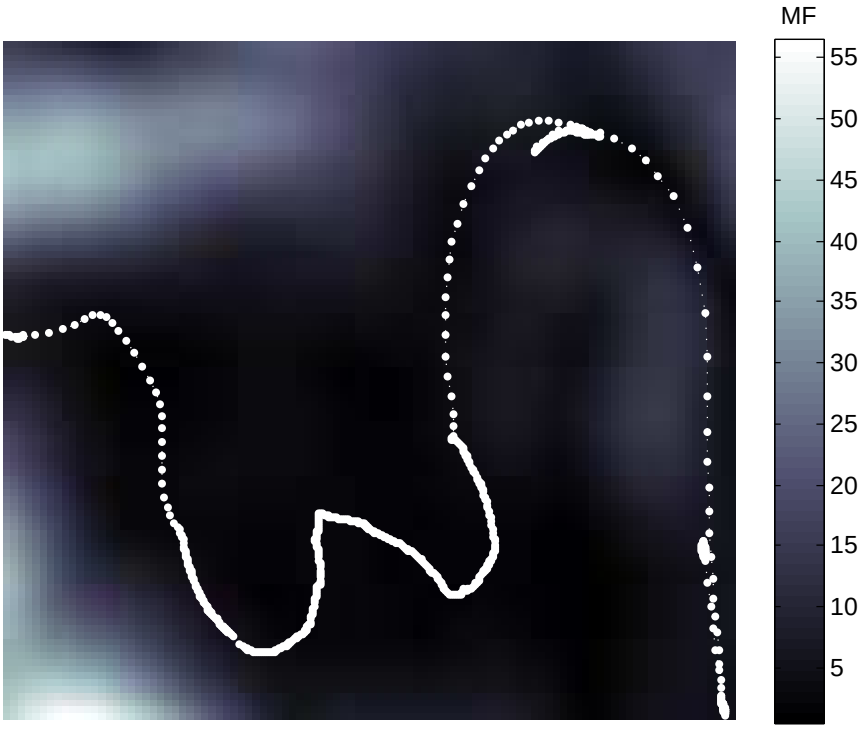


Fig. 6.1 Magnification Factor colormap computed over the GP-LVM latent space using the *Shuttle* dataset described in section § 5.3.1.

6.3 Computing geodesics

Given a latent space endowed with an expected Riemannian metric, we now consider how to compute geodesics (shortest paths) between given points. Once a geodesic is computed, its length can be evaluated through the numerical integration of Eq. 3.6.

An obvious solution to the shortest path problem is to discretise the latent space and compute shortest paths on the resulting graph using, e.g., Dijkstra's algorithm [Cormen et al., 1990]. The computational complexity of this approach, however, grows exponentially with the dimensionality of the latent space and the approach quickly becomes unfeasible. Moreover, this approach (presented in section § 6.3.1) will also introduce discretisation errors due to the finite size of the graph.

Instead, we propose solving the geodesic differential equation (3.7) numerically. This scales more gracefully as it only involves a discretisation of the geodesic curve which is always one-dimensional independently of the dimension of the latent space. This approach is presented in section § 6.3.2.

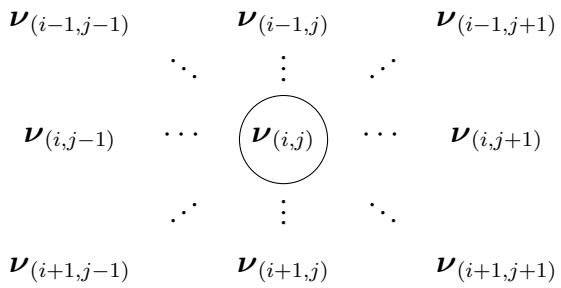
Remark

Note that the expectation of the GP-LVM metric tensor includes a covariance term depending on the covariance function of the GP prior. This implies that the metric tensor expands as the uncertainty over the mapping increases. Hence, curve length also increases when traversing uncertain regions and, as a consequence, geodesics will tend to avoid these regions. This nice effect comes from on the fact that in GP-LVM the covariance term depends on the values of the latent mapping. Not all the latent variable models have this property.

For instance, in GTM (c.f. Eq. 2.10) the covariance of the data posterior distribution is constant and equal to β^{-1}. It follows that the derivatives of the covariance terms with respect to the latent space are equal to zero, which means that also the covariance term which appears in the formulation of the expected metric in Eq. 6.3 is equal to zero. This observation shows that the formulation of the expected metric tensor for the GTM reduces to the original formulation of the metric given by Bishop et al. [1997a], which is computed without taking into account the covariance term. In the case of GTM, this omission makes no difference because the covariance of $\mathbf{J}$ is constant, due to the fact that the covariance of the conditional distribution given by Eq. 2.10 does not depend on the latent space.

6.3.1 Geodesics via discretisation

As a first approach, we propose [Tosi and Vellido, 2014] to discretise the space into a square grid of nodes $\boldsymbol{\nu}_{(i,j)} \in \mathbb{R}^q$ and build a weighted graph where every node is connected to its eight neighbours. A graphical visualisation of the connections of $\boldsymbol{\nu}_{(i,j)}$ is the following



The weights $\omega_{(i,j)}$ of the graph are computed according the local value of the MF

evaluated in the node $\boldsymbol{\nu}_{(i,j)}$

$$\omega_{(i,j)} = MF_{\mathbf{x}_*} = \sqrt{\det\left((\mathbf{W}\boldsymbol{\Psi}(\mathbf{x}_*))^\top \mathbf{W}\boldsymbol{\Psi}(\mathbf{x}_*)\right)}, \qquad \mathbf{x}_* = \boldsymbol{\nu}_{(i,j)}. \tag{6.8}$$

Notice that the MF expression presented in Eq. 4.12 for the t-GTM can be used for the computation of the MF in GTM by replacing the derivatives of the Student-t functions with the derivatives of the Gaussians.

The distance between any two nodes is then defined as the shortest path over the graph using Dijkstra's algorithm[1] [Cormen et al., 1990].

We display an illustrative example using 3-D data points sampled from a spiral. A standard GTM is trained wit it in order to learn a 2-D representation of the given data (see Fig. 6.2). We show in Fig. 6.3 that geodesic distances computed as shortest paths over the graph provide a more faithful interpolation between data points: in fact, the resulting interpolating path follows the natural structure of the data, giving a more faithful distance than to the Euclidean straight line.

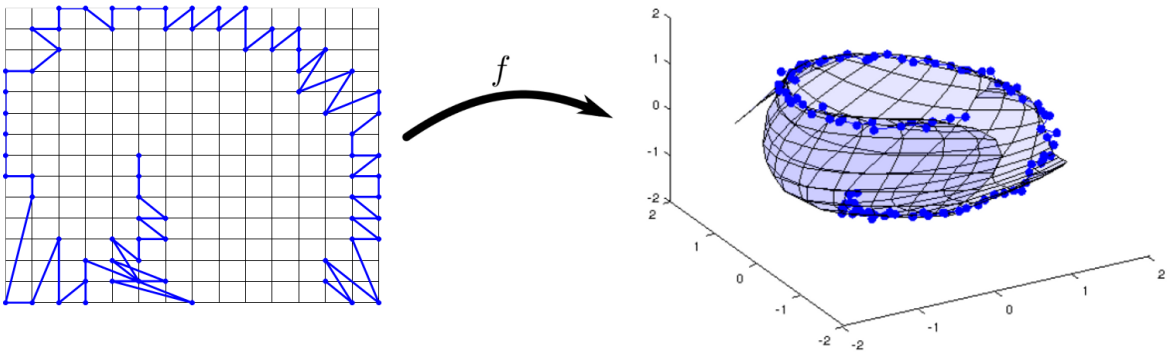


Fig. 6.2 A 3-D artificial dataset (a spiral) is used for training. The Generative Topographic Mapping (GTM) is used here as an illustrative technique; the model is trained over a 15×15 grid of nodes, laying on a 2-D latent space (left); training data are represented as blue dots, connected by a continuous line. The GTM latent grid is projected into the 3-D observed space to visualise the embed of the observations.

The computational complexity of the algorithm is that of the Dijkstra algorithm. To achieve a better accuracy in the geodesic computation, we can consider a finer grid, but this results in a growing computational cost.

[1]The algorithm has been implemented using the function graphshortest in Matlab®.

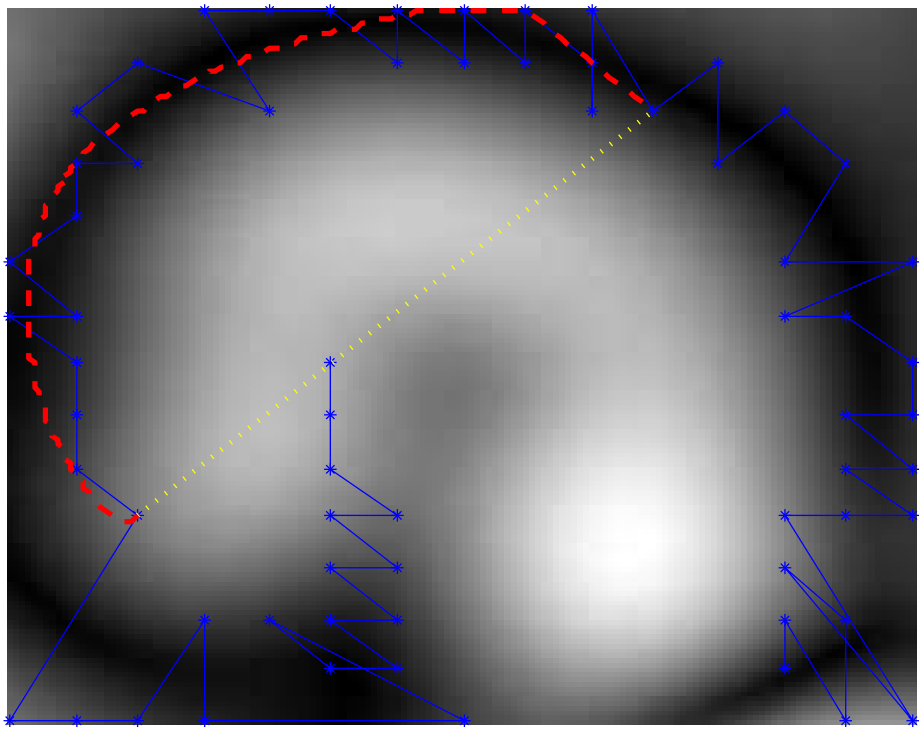


Fig. 6.3 Using the dataset given in Fig. 6.2, the local metric tensor is computed over the 2-D latent space and visualized via a Magnification Factor (MF) that quantifies distortion; the colormap shows areas of higher distortion in white and areas of lower distortion in black. Training points are represented in blue; the yellow dashed-dotted line represents the Euclidean distance between two training points; the red dashed line represents the geodesic distance between two training points, computed using the MF graph-based distance.

6.3.2 Geodesics via ODE's solution

Considering the concepts of differential geometry introduced in section § 3.2, we now focus on the definition of the geodesic curve given by Eq. (3.7). Here, the 2^{nd} order ODE can be rewritten in a standard way as a system of 1^{st} order ODE's, which can be solved using a four-stage implicit Runge-Kutta method[Kierzenka and Shampine, 2001][2]. This yields a smooth solution which is fifth order accurate. Alternatively, such equations can be solved by repeated GP regression [Hennig and Hauberg, 2014].

To evaluate Eq. 3.7, we need the derivative of the expected metric:

$$\frac{\partial \operatorname{vec} \mathbb{E}[\mathbf{G}(\mathbf{x})]}{\partial \mathbf{x}} = \frac{\partial \operatorname{vec}(\mathbb{E}[\mathbf{J}^\top]\mathbb{E}[\mathbf{J}] + p \cdot \operatorname{cov}(\mathbf{J}, \mathbf{J}))}{\partial \mathbf{x}}. \tag{6.9}$$

For the GP-LVM, this reduces to computing the derivatives of the covariance func-

[2]We use an off-the-shelf numerical solver (bvp5c in Matlab®); runnig times and computational cost are provided in the reference.

tion k (cf. Eq.(6.4)). Given two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^q$, a widely used covariance function is the *exponentiated quadratic* (EQ) (or *RBF*) kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \alpha \, \exp\left(-\frac{\omega}{2} \parallel \mathbf{x}_1 - \mathbf{x}_2 \parallel_2^2\right). \tag{6.10}$$

We choose here the *EQ* as an illustrative example, but our approach would apply to any other kernel that leads to a differential mapping.

This function is differentiable in $\mathbf{x}$ and will be used here (and in section § 6.4) to provide a specific algorithm. Eq. 2.37 and 2.38 are explicitly computed for the squared exponential kernel to have the explicit form of Eq. 6.4:

$$\left(\partial \mathbf{K}_{\mathbf{f}_1,\mathbf{f}_2}\right)_{1,j} = -\omega(x_1^{(j)} - x_2^{(j)})\, k(\mathbf{x}_1, \mathbf{x}_2) \tag{6.11}$$

$$\left(\partial^2 \mathbf{K}_{\mathbf{f}_1,\mathbf{f}_2}\right)_{i,l} = \tag{6.12}$$

$$= \begin{cases} \omega(x_1^{(i)} - x_2^{(i)})(x_1^{(l)} - x_2^{(l)})\, k(\mathbf{x}_1, \mathbf{x}_2), & i \neq l \\ \omega(\omega(x_1^{(i)} - x_2^{(i)})^2 - 1)\, k(\mathbf{x}_1, \mathbf{x}_2), & i = l \end{cases}$$

Due to symmetry conditions, the upper triangular of the Hessian matrix is sufficient to the computation. Note that, for our choice of kernel, the Hessian is diagonal and constant for $\mathbf{x}_1 = \mathbf{x}_2$, which is the case of $\partial^2 \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*}$, so there is no need to compute its derivative (which appears in the expression of $\partial \operatorname{vec} \mathbf{G}$).

6.4 Experiments and results

In the following section we compute geodesics via ODE's solution with GP-LVM over three different datasets.

6.4.1 Motivating example: Images of handwritten digits

When the mappings $f_j(\cdot)$ are nonlinear, the LVM can potentially capture non-linearities on the data and thereby provide an even lower dimensional representation as well as a more useful view of the data. While this line of thinking is popular, it is not without its practical issues. As an illustrative example, Fig. 6.4 shows the latent representation of a set of artificially rotated images obtained through a GP-LVM. The dataset consists a single image of a hand-written digit (number 5) rotated from 0 to 360 degrees to produce 200 rotated images.

It is clear from the display that the latent representation captures the underlying periodic structure of the process which generated the data (a rotation). If we want

to analyse the data in the latent space, e.g. by interpolating latent points, our current tools are insufficient. As can be seen, fitting a straight line in the latent space between the two-points leads to a solution that does not interpolate well in the data space: the interpolant goes through regions where the data does not reside, regions where the actual functions, $f_j(\cdot)$, cannot be well determined.

We then estimate a GP-LVM model[3] with a $q = 2$ dimensional latent space; the latent space is shown in Fig. 6.4. We interpolate two points using either a straight line or a geodesic, and reconstruct images along these paths. The results in Fig. 6.5 show the poor reconstruction of the straight-line interpolator. The core problem with this interpolator is that it goes through regions with little data support, meaning that the resulting reconstruction will be similar to the average of the entire data set.

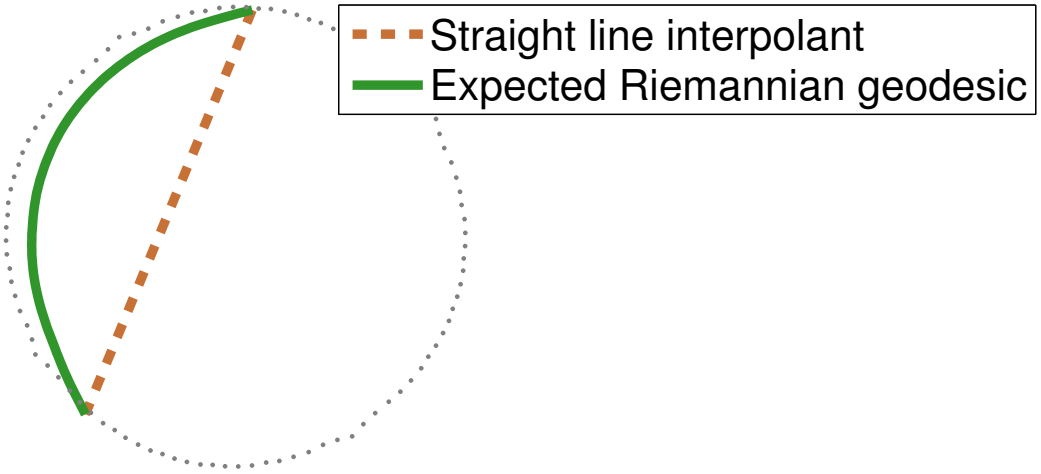


Fig. 6.4 The latent space from a GP-LVM that was trained over a dataset of artificially rotated digits. Black dots represent the latent points. The dashed brown line show the commonly used straight-line interpolant, and the green curve is the suggested expected Riemannian geodesic. This figure is best viewed in colour.

In the next two sections, we consider experiments on real data, but the reported results are similar to those obtained in the synthetic digit experiment. First, we consider images of rotating objects (section § 6.4.2), and then motion capture data (section § 6.4.3).

[3]Software from the Machine Learning group, University of Sheffield http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/software.html

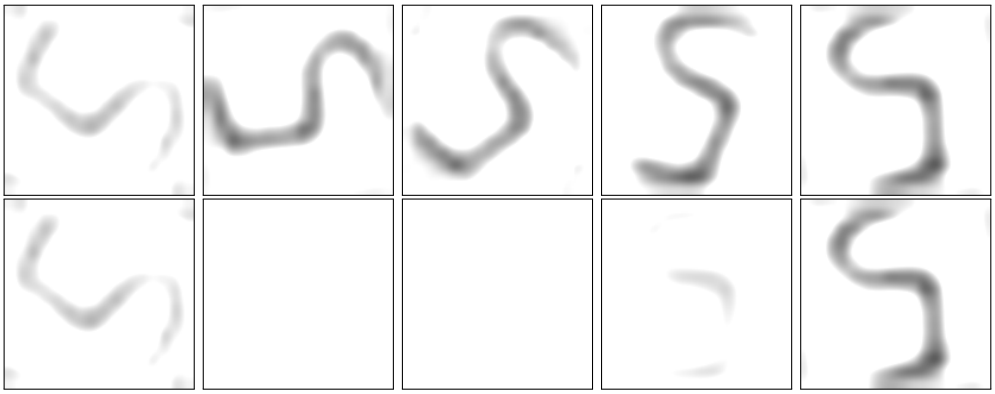


Fig. 6.5 Rotated digit sampled from the latent space displayed in Fig. 6.4. Inference after sampling over the latent space following the Geodesic distance (top row) and the Euclidean distance (bottom row). Images are inverted and bicubically upscaled for improved viewing.

6.4.2 Images of rotating objects

In this experiment, we consider images from the COIL data set [Nene et al., 1996], which consists of images from a fixed camera depicting 100 different objects on a motorised turntable against a black background. Each image is acquired after a 5-degree rotation of the turntable, giving a total of 72 images per object. Here, we only consider the images of object 74 (a rubber duck, Fig. 6.6) as an example, but similar results are attained for other objects.

Fig. 6.6 Objects from COIL 100 dataset. Object 74: a rubber duck (right).

A $q = 2$ dimensional latent space is estimated using the GP-LVM, and two latent points are interpolated using either a straight line according to the Euclidean distance, or a geodesic. Reconstructed images along the interpolated paths are shown in Fig. 6.7. It is clear that the geodesic gives a better interpolation, as it avoids regions with high uncertainty associated (which are unavoidably crossed if using the

Euclidean distance).

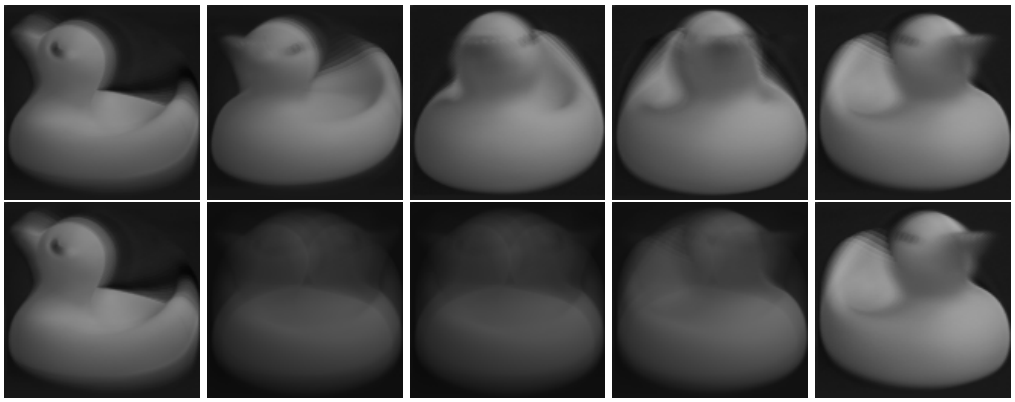


Fig. 6.7 COIL example image reconstruction. Inference after sampling over the latent space following the geodesic (top row) and the Euclidean straight line (bottom row).

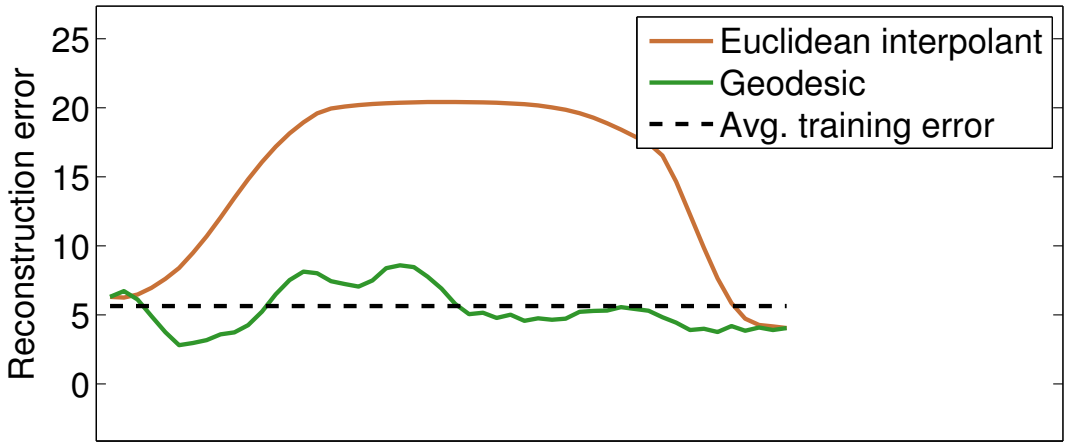


Fig. 6.8 COIL reconstruction error. Inference after sampling over the latent space following the geodesic (green) and the Euclidean straight line (brown). For reference, the average reconstruction error of the latent observations is shown as well (dashed). This figure is best viewed in colour.

To measure the quality of the different interpolators, we reconstruct 50 images equidistantly along each interpolating path and measure the distance to the nearest neighbour in the training data. This is shown in Fig. 6.8, which, for reference, also shows the average reconstruction error of the latent representations of the training data, or average training error (ATE)

$$\text{ATE} = \frac{1}{N}\sum_{n=1}^{N} \|\mathbb{E}\left[f(\mathbf{x}_n)\right] - \mathbf{y}_n\|. \tag{6.13}$$

It is clear that the straight line interpolator performs poorly away from the endpoints, while the geodesic yields errors which are comparable to the average error of the latent representation of the training data.

6.4.3 Human motion capture

We next consider human motion capture data from the *CMU Motion Capture Database*[4]. Specifically, we study motion 16 from subject 22, which is a repetitive *jumping jack* motion. Each time instance of this data consist of a human pose as acquiried by a marker-based motion capture system; see Fig. 6.12 for example data. We represent each pose by the three-dimensional joint positions, i.e. as a vector $\mathbf{y}_{n,:} \in \mathbb{R}^3$, where P denotes number of joint positions.

We estimate a GP-LVM using dynamics as is common for this type of data ([Wang et al., 2008], extended with further research by Damianou et al. [2011]). The dynamics constraints the latent space $\mathbf{X}$ to be smooth, by using a temporal prior. The resulting latent space is shown in Fig. 6.10, and the metric tensor is shown in Fig. 6.9, where the background colour is proportional to the magnification factor (Eq. 6.7) of the expected Riemannian metric.

As can be seen, the latent points $\mathbf{x}_{n,:}$ follow a periodic pattern as expected for this motion, and the metric tensor is generally smaller in regions of high data density.

We pick two latent extremal points of the motion ($\mathbf{x}_1$ and $\mathbf{x}_T$) and interpolate them using the Euclidean straight line and the expected Riemannian geodesic. Fig. 6.10 shows the interpolants: again, the geodesic follows the trend of the data while the straight line goes through regions with high model uncertainty. Reconstructed poses along the interpolants are shown in Figs. 6.13 and 6.14. A comparison with the intermediate poses ($\mathbf{x}_2 \dots \mathbf{x}_{T-1}$) in the training sequence (see Fig. 6.12) shows that the geodesic interpolant is a more truthful reconstruction compared to that of the straight line.

To measure the quality of the reconstruction, we note that the length of the subject's limbs should stay constant throughout the sequence. Our representation does, however, not enforce this constraint. Fig. 6.11 shows the length of the subjects forearm for the two reconstructions along with the correct length. The straight line interpolant drastically changes the limb lengths, while the geodesic matches the ground truth well. Similar observations have been made for other limbs.

[4] http://mocap.cs.cmu.edu/

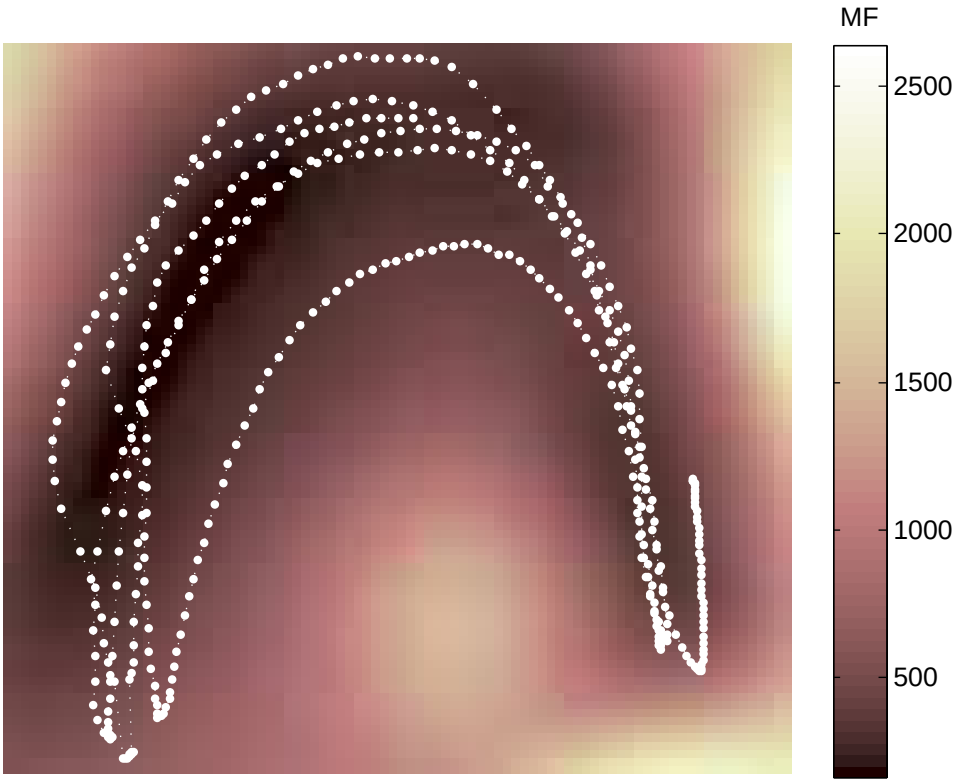


Fig. 6.9 GP-LVM latent space representation for the motion capture data. White dots denote latent points $\mathbf{x}_n$, whereas the background colour is proportional to the values of the MF (6.7), according to the code expressed by the colorbar. It is quite obvious that latent points are spread across paths of low MF, whereas latent space areas of high MF are avoided.

6.5 Discussion

When the mapping between a latent space and the observation space is not isometric (the common case for non-linear mappings), a Euclidean distance measure in the latent space does not match that of the original observation space. In fact, the distance measures in the latent and observation spaces can be arbitrarily different. This makes it difficult to perform any meaningful statistical operation directly in the latent space as the used metric is difficult to interpret.

We solve this issue by carrying the metric from the observation space into the latent space in the form of a *random Riemannian metric*. This gives a distribution over a smoothly changing local metric at each point in the latent space. We then provide an expression for the *expected* local metric and show how shortest paths (geodesics) can be computed numerically under the resulting metric. These geodesics provide natural generalisations of straight-lines and they are, as a result, suitable for interpolation under the new metric.

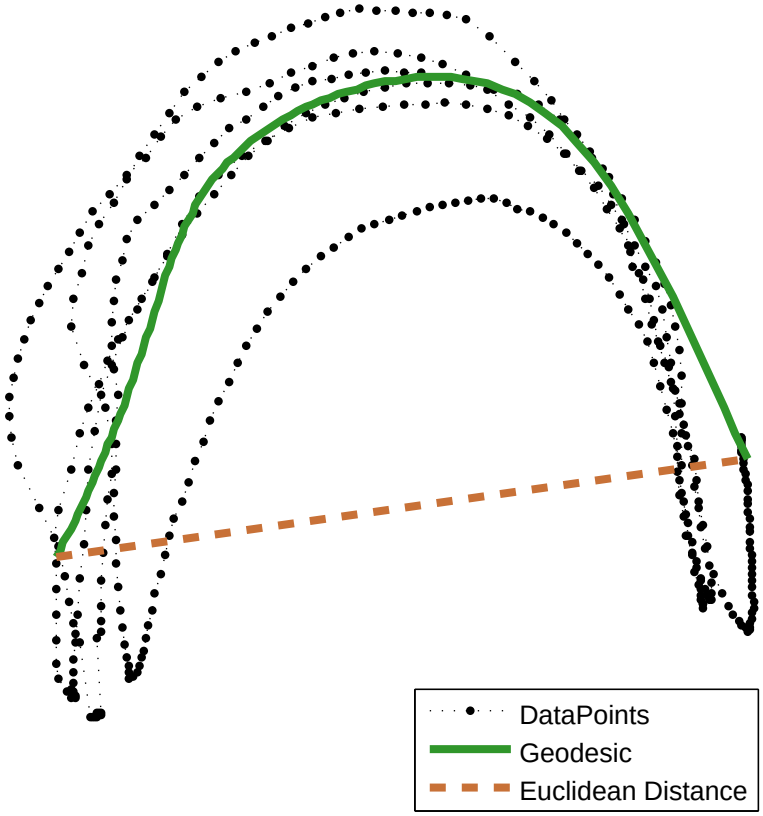


Fig. 6.10 GP-LVM latent space for the motion capture data. Black dots denote latent points $\mathbf{x}_n$. The green curve denotes the geodesic interpolant, while the dashed brown curve is the straight-line interpolant. This figure is best viewed in colour.

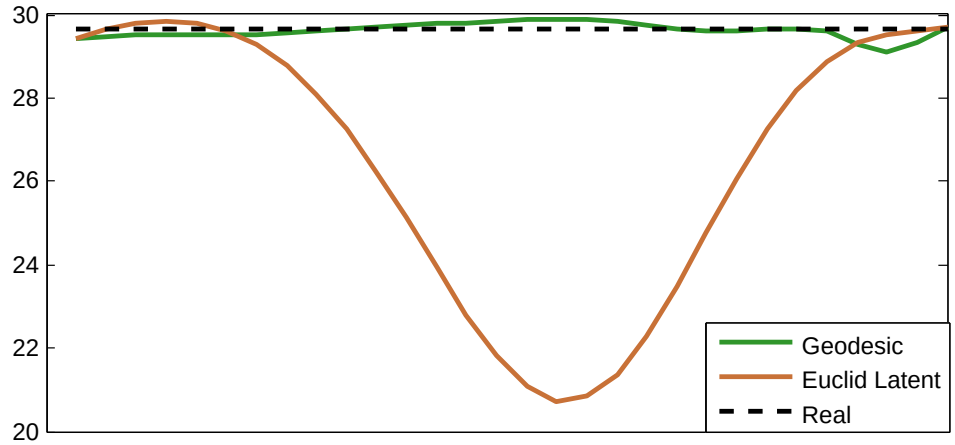


Fig. 6.11 Length, in centimetres, of the subjects forearm during latent space interpolation. The blue curve is according to the geodesic interpolant, and the red dashed curve is according to the straight-line interpolant. For reference, the black dots show the true length.

For the GP-LVM model, the expected metric depends on its uncertainty, in such

a way that distances become longer in regions of high uncertainty. This effectively forces geodesic curves to avoid uncertain regions in the latent space, which is the desired behaviour for most applications. It is worth noting that a similar analysis for the GTM does *not* provide a metric with this capacity as the uncertainty is constant in this model.

The idea of considering the expected metric is practical as it turns the latent space into a Riemannian manifold and this opens up to many applications. E.g. tracking can be performed in the latent space through a Riemannian Kalman filter [Hauberg et al., 2013], classification can be done using the geodesic distance, etc.

It is, however, potentially misleading to only consider the expectation of the metric rather than the entire distributions of metrics. Although, if the latent dimension is much lower than the observed data dimension, it can be shown that the distribution of the metric concentrates around its mean. But, in general, *random Riemannian manifolds* are mathematically less well-understood, e.g. it is known that geodesics are almost surely not length minimising curves under a random metric [LaGatta and Wehr, 2014]. We are suggesting that manifolds derived from data are necessarily uncertain, and there is much to gain from further consideration of these spaces, which then naturally lead to distributions over geodesics, distances, angles, curvature and so forth.

In this chapter, we have only considered how geometry can be used to understand an already estimated LVM, but it is also worth considering if this geometry can be used as part of the LVM estimation. That is, it would be worth investigating if a prior on the curvature of the latent manifold is an effective way to influence learning.

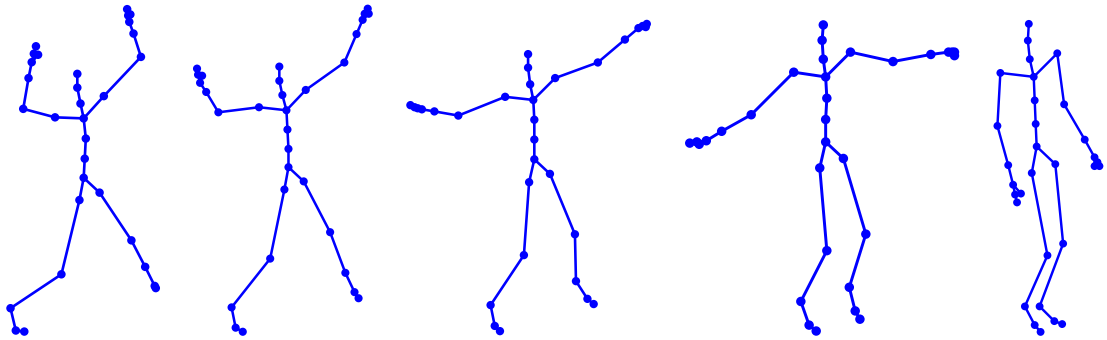


Fig. 6.12 Example poses from the motion capture data. These poses are temporarily spaced between the end-points of the interpolating curves, i.e. they are comparable to the interpolated reconstructions.

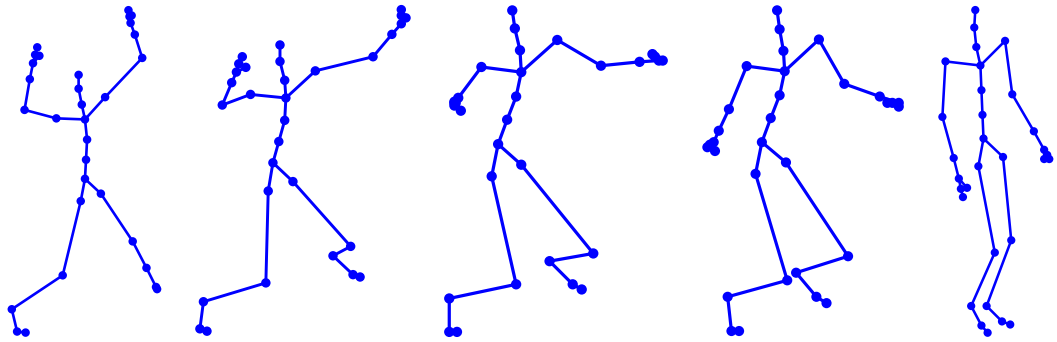


Fig. 6.13 Interpolated poses according to the straight-line interpolant. In particular, note the bending of the knees, which does not occur in the training data.

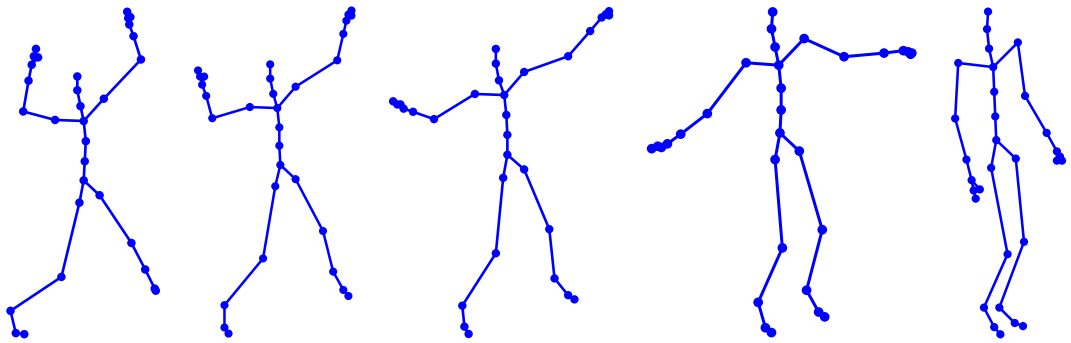


Fig. 6.14 Interpolated poses according to the geodesic. These are visually similar to the poses in Fig. 6.12.

Chapter 7

Conclusions

The final chapter of the thesis aims to summarise the main contributions presented in previous chapters and sketch some avenues for future research.

7.1 Summary of the thesis and its main contributions

In this thesis, we have addressed the problem of visualization of high-dimensional data sets, with the main objective of improving the interpretability (and as result, the usability) of the probabilistic non-linear dimensionality reduction models used to generate such visualization. To do so, we have mainly, but not only, exploited the intrinsic geometrical structure of the aforementioned models.

The main contribution in the domain of visualization techniques is given by the **Carotgram-based** representation, presented in chapter § 4. This novel technique, inspired by cartographic maps in the geography domain, has been used to reintroduce in the visualization space a loss of information in which the non-linear mapping incurs. The analytical quantification of such a distortion has been expressed in the form of Magnification Factors, and then computed and visualised together in the form of the Cartogram maps. Results for the case of Self Organizing Maps and Student-t Generative Topographic Mapping have been presented.

The research carried out in this thesis can be applied to multivariate data of different nature and from different domains. In chapter § 5 we presented experimental results for multivariate time series. We improved interpretability for the VB-GTM-TT model and we introduced the explicit estimation and visualisation of the cumulative probabilities of transition between states $CSTP$.

The main theoretical contribution in the domain of **Random Geometries** is

provided in the last chapter, § 6. There we reinterpreted latent variable models as Riemannian manifolds, by pulling back the Riemannian metric from the observation space. Differently from previous approaches to metric learning, our local metric is defined in a probabilistic way, providing an explicit expression of its probability distribution for the considered models. Experimental results have shown that inference made following the Riemannian metric leads to a more faithful generation of new data.

In addition, we have stated that the algorithms described in this thesis can be extended to other generative latent variable models characterised by a smooth mapping between the latent space and the observed space. This property points out to the **portability** of the proposed approach.

In conclusion, we have carried out an extensive analysis of the problem of interpretability in probabilistic dimensionality reduction, from the differential geometry point of view. We hope that this work will be of help to other researches with related focus and open the way to novel investigation, such as the one suggested in the next session.

7.2 Open questions and future directions

Random geometries

We have shown how to define a distribution over the metric in latent variable models. In particular, this was achieved by pulling back the metric from the observation space into the latent space in the form of a random Riemannian metric. This research opens to new streams of investigation in the field of *Random Geometries* in relation to machine learning. Random Riemannian manifolds are mathematically less well-understood than Riemannian manifolds. In fact, it is known that geodesics are almost surely not length minimising curves under a random metric [LaGatta and Wehr, 2014]. We are suggesting that manifolds derived from data are necessarily uncertain, and there is much to gain from further consideration of these spaces, which then naturally lead to distributions over geodesics, distances, angles, curvature and so forth.

In this thesis, we have only considered how probabilistic geometry can be used to understand an already estimated generative latent variable model. This work opens the way to promising direction of investigation, namely the applicability of probabilistic geometry as part of the model estimation itself and, if so, it is worth understanding its influence in the learning process.

Distributions of geodesics

We have developed a probabilistic framework where the support of the data can be interpreted as a random Riemannian manifold and geodesic distances can be computed by taking the uncertainty of the metric into account. The preliminary results presented in chapter § 6 give rise to some questions, such as: **(1)** How does uncertainty defined over the the metric tensor impact the geodesic distances computed in the observed space? **(2)** What are the conditions of existence and uniqueness of geodesics in a more general setting? **(3)** How do we analytically define, if exists, a distribution over geodesics? **(4)** What is the behaviour of the lengths of geodesics when the number of features grows? Do distances concentrate as the dimensionality of the feature space goes to infinity?

In order to answer these questions, we are currently investigating how to develop an algorithm to explicitly compute distributions over geodesics in probabilistic dimensionality reduction. One approach to be considered is that which entails combining the local distributions of the metric tensor for a given set of points by considering a joint sample: this way we would obtain samples of the metric along the whole manifold (i.e. samples from the distribution of the random manifold).

We display examples of these samples in Fig. 7.1, where the diagrams refer to the 3-dimensional visualization of the MF computed over a 2-dimensional GP-LVM latent space (the 3-D visualisation has been introduced in section § 3.3). Here the first diagram represents the plot of the MF of the expected metric tensor, defined according to Eq. 6.6; the rest of diagrams show some random joint samples from the Wishart distribution of the metric tensor (rather than showing just the mean).

Once the samples of the manifold are computed, we can compute (for each sample) the samples of geodesics using the algorithms proposed in section § 6.3.1 and section § 6.3.2. This is straightforward for the GP-LVM model and a Wishart distributed metric introduced above, and the results can be extended to similar generative models.

This will provide new theoretical insights into the problems of probabilistic geometries, as the conditions of existence and uniqueness of geodesics in a more general setting (i.e. wider class of models) are not well known. One challenge is the fact that, in general, the distribution of the geodesic can be very complex because even small changes in the metric tensor can result in a big change in the geodesic and its length. Moreover, if the dimensionality of the feature space is increased and sent to infinity, this is thought to have an effect on the lengths of interpolating paths, resulting in an effect of concentrated distances. This aspect represents one of the

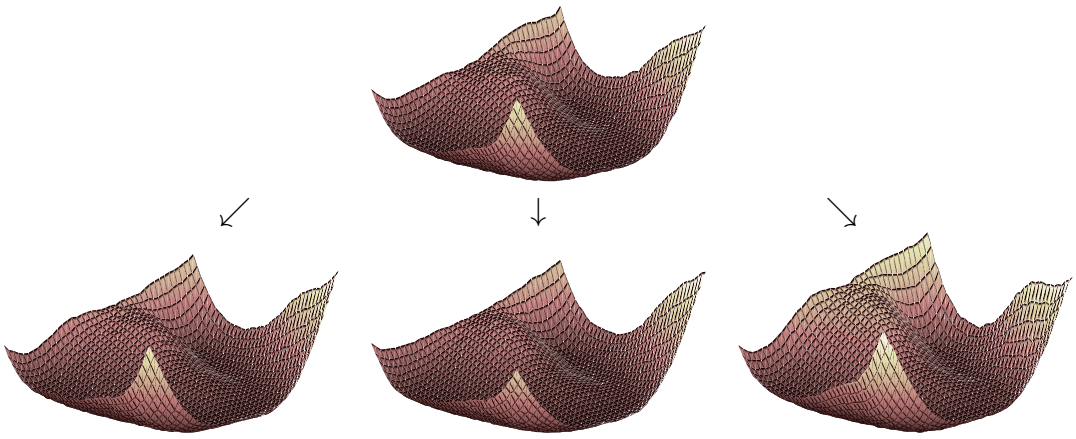


Fig. 7.1 On the top plot: mean of the distribution of a random manifold (generated after training a GP-LVM over a *jumping jacks* motion form the CMU database, c.f. Experimental results in section § 6.4.3). Three different joint random samples are generated from this Wishart, and values of the MF factors are represented by a colormap. A 3-D visualization of the MF is used: the vertical axes of each plot represents the values of the MF.

future directions of research.

Probabilistic numerics

New sparks of research aim to identify numerical methods as learning problems using probabilistic models. This approach is know as *Probabilistic numerics*[1] and addresses classical optimisation algorithms and numerical methods for the solution of differential equations and integrals.

Such probabilistic numerical methods, applied to find solutions of ordinary differential equations (ODEs), have an impact on the analysis of statistical Riemannian manifolds [Hennig and Hauberg, 2014]. In particular, the very recent work of Schober et al. [2014] provides a probabilistic model for the solution of ODEs which matches the classical Runge-Kutta method. A future direction of research is to investigate the combination of these recent advances in probabilistic numerics and the propagation of the uncertainty defined over the metric tensor through the geodesic ODE solver presented in section § 6.3.2.

Big data

In this thesis, we have addressed the issue of high-dimensional data as a problem of datasets with a high number of features. But we also need to consider that, given the ever increasing amount of available data generated on daily basis in different

[1] About: http://probabilistic-numerics.org/

domains, we have at our disposal a growing amount of datasets which are big mostly in terms of number of observations. This has come to be popularised under the name of *Big data* ant constitutes a very up-to date problem.

Performing inference with Gaussian processes-based techniques suffers from a high computational cost, and scaling up GPs is a topic of ongoing research. The complexity can be reduced using appropriated approximation techniques as well as appropriate distributed algorithms. Very recent results [Hensman et al., 2013; Gal et al., 2014] apply GPs to data of the order of $N \sim 10^6$. Following these promising results, we aim to extend the approach presented in this thesis to the analysis of larger datasets.

Intrinsic dimensionality of the dataset

While performing dimensionality reduction, we have mostly set the dimensionality of the latent space to be $q = 2$, in order to display the experimental results on the visualisation space. This has been done because one of the scopes of this thesis is to make progress in the visualization techniques in order to improve the interpretability of the considered models. The theoretical results presented, however, are valid for any choice of latent dimension $q \leq p$.

By removing the restriction of a 2-D latent space, we face the problem of how to set the value of q. This question has been answered for a certain class of models, and in the context of GP based dimensionality reduction has been solved by the Variational GP-LVM [Titsias and Lawrence, 2010; Damianou et al., 2014]. In this model the algorithm is capable of computing the intrinsic dimensionality of the data by optimising the lengthscales of the kernel in each latent dimension (and, eventually, switching off the non relevant ones).

Extension to other models

The focus of this work is mainly on two models of interest, the Generative Topographic Mapping and Gaussian Process Latent Variable Model, both part of the family of generative latent variable models. We have explicitly defined a distribution over the local metric and we have visualised such metric using the values of Magnification Factors. The conditions to extend our approach to a wider class of probabilistic models is to have a smooth mapping between the latent space and the observed space. This is the case of GP-based models that use differentiable kernel functions, which opens to a wide class of models. Examples of a straightforward extension have been done in this thesis using the GP dynamical system (GPDM,

[Wang et al., 2008; Damianou et al., 2011]). Further extensions are object of ongoing investigation, using the Variational GP [Titsias and Lawrence, 2010; Damianou et al., 2014], and the more recent Deep GP [Damianou and Lawrence, 2013]. These models apply a fully Bayesian approach

Appendix A

Mathematical Background

This appendix aims to make the thesis self-contained, providing a short reference to the basic mathematical identities used among the document.

A.1 Gaussian Identities

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of random variables. Let's consider two Gaussian random vectors

$$\mathbf{x}_A \sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A) \quad \text{and} \quad \mathbf{x}_B \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B).$$

The **joint distribution** $p(\mathbf{x}_A, \mathbf{x}_B)$ is given by

$$\begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_A & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{AB}^\top & \boldsymbol{\Sigma}_B \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix}, \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_A & \tilde{\boldsymbol{\Sigma}}_{AB} \\ \tilde{\boldsymbol{\Sigma}}_{AB}^\top & \tilde{\boldsymbol{\Sigma}}_B \end{bmatrix}^{-1} \right). \tag{A.1}$$

the **marginal distribution** is given by

$$\begin{aligned} p(\mathbf{x}_A) &= \textstyle\int p(\mathbf{x}_A, \mathbf{x}_B) d\mathbf{x}_B = \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A) \\ p(\mathbf{x}_B) &= \textstyle\int p(\mathbf{x}_A, \mathbf{x}_B) d\mathbf{x}_A = \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B). \end{aligned} \tag{A.2}$$

The **conditional distribution** of $\mathbf{x}_A$ given $\mathbf{x}_B$ is

$$p(\mathbf{x}_A \mid \mathbf{x}_B) = \mathcal{N}\left(\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B), \boldsymbol{\Sigma}_A - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_B^{-1}\boldsymbol{\Sigma}_{AB}^\top\right), \tag{A.3}$$

similarly the conditional distribution of $\mathbf{x}_B$ given $\mathbf{x}_A$ is

$$p(\mathbf{x}_B \mid \mathbf{x}_A) = \mathcal{N}\left(\boldsymbol{\mu}_B + \boldsymbol{\Sigma}_{AB}^\top\boldsymbol{\Sigma}_A^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A), \boldsymbol{\Sigma}_B - \boldsymbol{\Sigma}_{AB}^\top\boldsymbol{\Sigma}_A^{-1}\boldsymbol{\Sigma}_{AB}\right). \tag{A.4}$$

The **product** of two Gaussian distributions over the same domain gives an *unnormalised* Gaussian

$$\begin{aligned} &\mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A)\mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B) \propto \mathcal{N}(\boldsymbol{\mu}_C, \boldsymbol{\Sigma}_C) \\ &\boldsymbol{\mu}_C = \boldsymbol{\Sigma}_C \left(\boldsymbol{\Sigma}_A^{-1}\boldsymbol{\mu}_A - \boldsymbol{\Sigma}_B^{-1}\boldsymbol{\mu}_B\right)^{-1} \\ &\boldsymbol{\Sigma}_C = \left(\boldsymbol{\Sigma}_A^{-1} - \boldsymbol{\Sigma}_B^{-1}\right)^{-1} \end{aligned} \tag{A.5}$$

A.2 Matrix identities

A.2.1 Matrix inversion lemma (Woodbury matrix identity)

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1} \tag{A.6}$$

A.2.2 Matrix determinant lemma

$$\mid A + UWV^{\top} \mid = \mid W^{-1} + V\top A^{-1}U \mid \mid W \mid \mid A \mid . \tag{A.7}$$

A.3 The diffusion equation

The diffusion equation (also known as the Heat equation), describes a system where a quantity u is considered as a function of the spatio-temporal coordinates. In the classical problem this function $u(\mathbf{x}, t)$ is taken to be the temperature of the particles in a conductive body. Given the initial condition at $t = 0$, the equation describes the values of the temperature for $t \in [0, \infty)$.

$$\alpha\nabla^2 u - \frac{\partial u}{\partial t} = 0, \tag{A.8}$$

References

S. Amari and H. Nagaoka. *Methods of information geometry*. Translations of mathematical monographs; v. 191. American Mathematical Society, 2000.

T. W. Anderson. The non-central Wishart distribution and certain problems of multivariate statistics. *The Annals of Mathematical Statistics*, 17(4):409–431, Dec. 1946.

M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304–1330, 2007.

D. J. Bartholomew. *Latent Variable Models and Factor Analysis*. Charles Grifn& Co. Ltd, London, 1987.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

C. Bishop, M. Svensén, and C. K. I. Williams. Magnification factors for the SOM and GTM algorithms. In *Workshop on Self-Organizing Maps, WSOM 2014*, Advances in Intelligent Systems and Computing. Springer, 1997a.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

C. M. Bishop, G. E. Hinton, and I. G. D. Strachan. GTM through time. In *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K.*, page 111–116, January 1997b.

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998a.

C. M. Bishop, M. Svensén, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1-3):203–224, 1998b.

C. Bregler and S. M. Omohundro. Nonlinear image interpolation using manifold learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 973–980. MIT Press, 1994.

T. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Cambridge, MA, 1990.

A. Damianou and N. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS-13)*, AISTATS '13, pages 207–215. JMLR WCP 31, 2013.

A. C. Damianou, M. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2510–2518. 2011.

A. C. Damianou, M. K., Titsias, and N. D. Lawrence. Variational inference for uncertainty on the inputs of Gaussian process models. *arXiv preprint arXiv:1409.2287*, 2014.

M. P. do Carmo. *Riemannian Geometry*. Birkhäuser Boston, January 1992.

Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Rev.*, 41(4):637–676, Dec. 1999.

T.-C. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

Y. Gal, M. van der Wilk, and C. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems (NIPS)*. In press, 2014.

D. L. García, Nebot, and A. Vellido. Telecommunications customers churn monitoring using flow maps and cartogram visualization. In S. Coquillart, C. Andújar, R. S. Laramee, A. Kerren, and J. Braz, editors, *GRAPP/IVAPP*, pages 451–460. SciTePress, 2013.

M. Gastner and M. Newman. Diffusion-based method for producing density-equalizing maps. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 101 (20), pages pp.7499–7504. National Academy of Sciences, 2004.

C. F. Gauss. Disquisitiones generales circa superficies curvas. *Commentationes Societatis Regiae Scientiarum Gottingesis Recentiores*, VI:99–146, 1827.

N. Gianniotis. Interpretable magnification factors for topographic maps of high dimensional and structured data. In *Symposium on computational intelligence and data mining (CIDM)*, pages 238–245. IEEE, 2013.

B. Hammer, A. Hasenfuss, and T. Villmann. Magnification control for batch neural gas. *Neurocomputing*, 70(7-9):1225–1234, 2007.

S. Hauberg, O. Freifeld, and M. Black. A geometric take on metric learning. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2033–2041. MIT Press, 2012.

S. Hauberg, F. Lauze, and K. S. Pedersen. Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 46(1):103–120, May 2013.

P. Hennig and S. Hauberg. Probabilistic solutions to differential equations and their application to Riemannian statistics. In *Proceedings of the 17th international Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 33, 2014.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Education and Psycology*, 24:414–441, 1933.

J. Kierzenka and L. F. Shampine. A BVP solver based on residual control and the Matlab PSE. *ACM Transactions on Mathematical Software*, 27(3):299–316, 2001.

T. Kohonen. *Self-organizing maps*, volume 3 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, December 2001.

T. LaGatta and J. Wehr. Geodesics of random Riemannian metrics. *Communications in Mathematical Physics*, 327(1):181–241, 2014.

N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of machine learning research*, 6:1783–1816, 2005.

N. D. Lawrence. A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *Journal of Machine Learning Research*, 13: 1609–1638, 2012.

J. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Information Science and Statistics, Springer, 2007.

M. Lewandowski, J. M. del Rincón, D. Makris, and J.-C. Nebel. Temporal extension of Laplacian eigenmaps for unsupervised dimensionality reduction of time series. In *ICPR*, pages 161–164. IEEE, 2010.

J. Lin, M. Vlachos, E. J. Keogh, and D. Gunopulos. Iterative incremental clustering of time series. In *EDBT*, volume 2992 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2004.

K. Mardia, J. Kent, and J. Bibby. *Multivariate analysis*. Academic Press; New York, 1979.

V. Marx. Biology: The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

F. Mulier and V. Cherkassky. Self-organization as an iterative kernel smoothing process. *Neural Computation*, 7(6):1165–1177, 1995.

S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, Feb 1996.

I. Olier. *Variational Bayesian Algorithms for the Generative Topographic Mapping and its Extensions*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2008.

I. Olier and A. Vellido. A variational formulation for GTM through time. In *International Joint Conference on Neural Networks, IJCNN*, pages 516–521. IEEE, 2008a.

I. Olier and A. Vellido. Variational Bayesian generative topographic mapping. *Journal of Mathematical Modelling and Algorithms*, 7(4):371–387, 2008b.

I. Olier and A. Vellido. Advances in clustering and visualization of time series using GTM through time. *Neural Networks*, 21(7):904–913, 2008c.

K. Pearson. On lines and planes of closest fit to points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

D. Peel and G. J. Mclachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 2000.

J. Pointer. The cortical magnification factor and photopic vision. *Biological Reviews*, 61(2):97–119, 1986.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipies In C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1988.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of IEEE*, volume 77, pages 257–286. IEEE, 1989.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA, 2006.

B. Riemann. *On the Hypotheses Which Lie at the Foundations of Geometry*. 1854.

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 626–632. The MIT Press, 1997.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

M. Schober, D. Duvenaud, and P. Hennig. Probabilistic ODE solvers with Runge-Kutta means. In *Advances in Neural Information Processing Systems (NIPS)*. In press, 2014.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. In *Proceedings 1997 International Conference on Artificial Neural Networks, ICANN'97*, page 583, Lausanne, Switzerland, 1997.

E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen. Derivative observations in Gaussian process models of dynamic systems. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1033–1040. MIT Press, 2002.

M. Svensén. *GTM: The Generative Topographic Mapping* . PhD thesis, Aston University, 1998.

J. B. Tenenbaum. Mapping a manifold of perceptual observations. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems (NIPS)*. The MIT Press, 1997.

J. B. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

M. Tenenbaum and H. Pollard. *Ordinary Differential Equations*. Dover Publications, 1963.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 6(3):611–622, 1999.

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the 13th international Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 844–851, 2010.

W. R. Tobler. Thirty-five years of computer cartograms. *Annals of the Association of American Geographers*, 94:58–73, 2004.

A. Tosi and A. Vellido. Cartogram representation of the batch-SOM magnification factor. In *The 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 203–208, Bruges, Belgium, 2012.

A. Tosi and A. Vellido. Robust cartogram visualization of outliers in manifold learning. In *The 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 555–560, Bruges, Belgium, 2013.

A. Tosi and A. Vellido. Local metric and graph based distance for probabilistic dimensionality reduction. In *Proceedings of the Workshop on Features and Structures (FEAST 2014) International Conference on Pattern Recogni- tion (ICPR 2014)*, Stockholm, Sweden, 2014.

A. Tosi, S. Hauberg, A. Vellido, and N. D. Lawrence. Metrics for probabilistic geometries. In J. T. Nevin L. Zhang, editor, *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 800–808, Quebec City, Canada, 2014a. AUAI Press Corvallis, Oregon.

A. Tosi, I. Olier, and A. Vellido. Probability ridges and distortion flows: Visualizing multivariate time series using a variational Bayesian manifold learning method. In *Advances in Self-Organizing Maps, the 10th International (WSOM)*, Advances in Intelligent Systems and Computing, pages 55–64. Springer, 2014b.

A. Ultsch. U*-matrix: a tool to visualize clusters in high dimensional data. Technical Report 36, Philipps-University Marburg, Germany, 2003.

A. Vellido. Assessment of an unsupervised feature selection method for generative topographic mapping. In S. D. Kollias, A. Stafylopatis, W. Duch, and E. Oja, editors, *The International Conference on Artificial Neural Networks*, volume 4132 of *Lecture Notes in Computer Science*, pages 361–370. Springer, 2006a.

A. Vellido. Missing data imputation through GTM as a mixture of t-distributions. *Neural Networks*, 19(10):1624–1635, 2006b.

A. Vellido, W. El-Deredy, and P. J. G. Lisboa. Selective smoothing of the generative topographic mapping. *IEEE Transactions on Neural Networks*, 14(4):847–852, 2003.

A. Vellido, P. J. G. Lisboa, and D. Vicente. Robust analysis of MRS brain tumour data using t-GTM. *Neurocomputing*, 69(7-9):754–768, 2006.

A. Vellido, J. D. Martín, F. Rossi, and P. J. G. Lisboa. Seeing is believing: The importance of visualization in real-world machine learning applications. In *The 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2011.

A. Vellido, J. D. Martín, and P. J. G. Lisboa. Making machine learning models interpretable. In *The 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 163–172, 2012.

A. Vellido, D. L. García, and Nebot. Cartogram visualization for nonlinear manifold learning models. *Data Mininig and Knowledge Discovery*, 27(1):22–54, 2013.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 30(2):283–298, Feb. 2008.

H. Wässle, U. Grünert, J. Röhrenbeck, and B. Boycott. Retinal ganglion cell density and cortical magnification factor in the primate. *Vision Research*, 30(11):1897–1911, 1990.

K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *The Conferece for the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1683–1686. AAAI Press, 2006.