



UNIVERSITAT DE BARCELONA



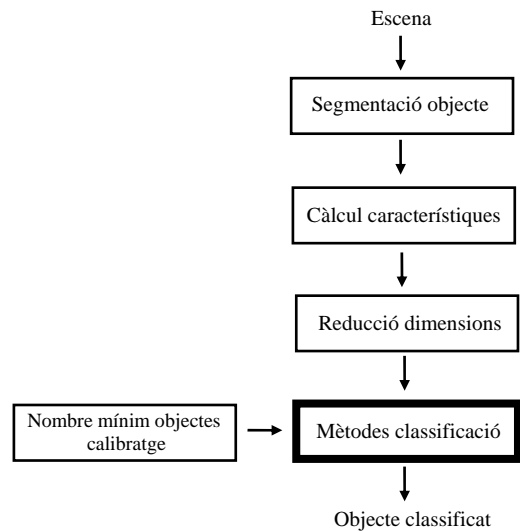
Departament de Física Aplicada i Òptica
Programa de Micro i Optoelectrònica Física
Bienni 1994-96

DISSENY D'UN PROTOCOL NUMÈRIC PER A LA
CLASSIFICACIÓ INVARIANT D'IMATGES APLICANT
TÈCNIQUES MULTIVARIANTS

Memòria presentada per optar al títol de doctor en Ciències Físiques

Directors:
Dr. Arturo Carnicer González
Dr. Ignacio Juvells Prades

Jordi-Roger Riba Ruíz
Barcelona, maig de 2000



5. Mètodes de classificació

En aquest capítol se suposa que les m característiques calculades per a cada objecte han estat transformades a un nombre m^* més reduït de variables. Si, pel motiu que fos, no s'hagués fet aquest pas de reducció de dimensions, en tots els raonaments que es fan a continuació, on digui m^* variables s'ha d'entendre m característiques.

Una vegada es tenen calculades les variables que serveixen per descriure cada objecte, es requereixen uns algorismes matemàtics que permetin classificar els objectes en alguna de les classes predefinides. Aquest procés de classificació es pot dur a terme utilitzant, d'una banda, mètodes multivariants de regressió o predicció (utilitzats com a classificadors), i de l'altra, mètodes pròpiament dits de classificació.

Com que en els problemes de visió artificial generalment es disposa d'objectes de calibratge, en aquest capítol es fixa només l'atenció en els algorismes supervisats de predicció i classificació. Això significa que forçosament s'ha de disposar d'un conjunt d'objectes la classe dels quals és coneguda. Aquests formen el conjunt de calibratge del model i serviran per determinar el valor dels paràmetres necessaris per establir la relació entre la matriu Y de sortides (graus de pertinença) i la matriu X de variables d'entrada.

També es disposarà d'un conjunt de test format per una col·lecció d'objectes la classe dels quals és desconeguda. El model matemàtic de classificació ha de classificar adequadament els objectes de test.

5.1. Mètodes de predicció o regressió

Els mètodes experimentals de predicció o de regressió serveixen per modelar dades mesurades, trobades experimentalment. Permeten establir una relació matemàtica entre les dades d'entrada (matriu X) i les de sortida (matriu Y). La figura 5.1. mostra l'afirmació anterior. Aquesta relació matemàtica serà més o menys complexa depenent del grau de complexitat de les dades experimentals i del model de regressió elegit. L'elecció del model és un punt clau del procés de caracterització de la relació entre les dades d'entrada i les de sortida.

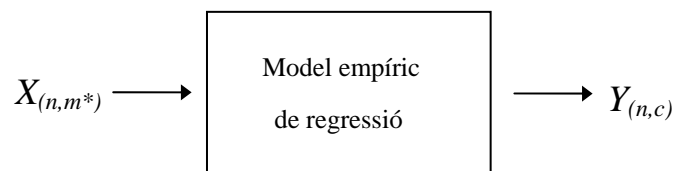


Figura 5.1.

En aquest treball, les entrades corresponen a cada una de les m^* variables calculades per a cada objecte. La variable j -èssima de l'objecte i -èssim es representa, per x_{ij} , amb $i = 1, 2, \dots, n$ i $j = 1, 2, \dots, m^*$. El conjunt $\{x_{ij}\}$ forma la matriu $X_{(n,m^*)}$.

Les variables de sortida (respostes de l'algorisme de predicció) representen el grau de pertinença de cada objecte respecte de cada una de les c classes d'objectes. La variable y_{il} simbolitza el grau amb què l'objecte i -èssim pertany a la classe l -èssima, amb $i = 1, 2, \dots, n$ i $l = 1, 2, \dots, c$. És freqüent realitzar la normalització següent:

$$y_{il} \in (0, 1)$$

El valor 0 indica que l'objecte i -èssim no pertany amb tota seguretat a la classe l -èssima, mentre que un valor 1 indica que hi pertany amb tota seguretat.

Els mètodes multivariants de regressió i les xarxes neuronals són exemples de mètodes de predicció. Els mètodes de regressió, a diferència dels mètodes de classificació, no associen directament un objecte a una classe, sinó que es limiten a donar un valor de sortida, generalment pertanyent a l'interval $(0, 1)$. Els models de regressió es poden classificar de les maneres següents:

Models univariables i multivariables

Quan cada variable dependent de sortida y_j només està relacionada amb una variable independent x d'entrada, el model és univariable. En canvi, quan cada variable y_j de sortida depèn de diverses variables x_i d'entrada, el model és multivariable.

Exemple de model univariable:

$$y_j = w_j \cdot f(x) + w_{0j} \quad (5.1)$$

Exemple de model multivariable:

$$y_j = \sum_{i=1}^m w_{ij} \cdot f(x_i) + w_{0j} \quad (5.2)$$

Models lineals i no lineals

En els models lineals s'estableix una relació de proporcionalitat entre les variables d'entrada i les de sortida, del tipus:

$$y_j = \sum_{i=1}^{m^*} w_{ij} \cdot x_i + w_{0j} \quad (5.3)$$

Els w_{ij} són els paràmetres desconeguts, el valor dels quals serà determinat durant el procés de calibratge del model.

En els models no lineals la relació sortida-entrada és del tipus:

$$y_j = \sum_{i=1}^{m^*} w_{ij} \cdot f(x_i) + w_{0j} \quad (5.4)$$

En la fórmula anterior, $f(x_i)$ representa una funció no lineal, el valor de la qual depèn del model escollit. Els w_{ij} són els paràmetres desconeguts, el valor dels quals serà determinat durant el procés de calibratge del model.

Models paramètrics i no paramètrics

Un model paramètric és aquell en el qual es coneix explícitament la relació matemàtica entre les variables dependents de sortida y_j i les variables independents d'entrada x_i . Aquesta relació pot ser, per exemple, de tipus lineal, polinòmica, exponencial, etc. En aquests tipus de models cal determinar un nombre normalment reduït però ben definit de paràmetres, el valor dels quals s'ajusta en el procés de calibratge del model. En canvi, en els models de regressió no paramètrics no hi ha un coneixement *a priori* del tipus de relació existent entre les variables y_j de sortida i les variables x_i d'entrada.

Les xarxes neuronals (tant les de tipus *Backpropagation* com les de tipus *Radial Basis*) poden ser considerades mètodes de regressió multivariable, no lineals i no paramètrics. En canvi, els models de regressió lineal múltiple (MLR, PCR, PLS, ...) poden ser considerats regressions multivariables, lineals i paramètriques.

5.1.1. Interpolació i generalització dels models de regressió

Els models empírics de predicció o regressió poden ser interpretats com a problemes d'interpolació. Matemàticament, la **interpolació** es defineix així:

Donat un conjunt de n vectors diferents $\check{x}_i = \{x_{ij}\}$ de \mathfrak{R}^{m^*} , amb $i = 1, 2, \dots, n$ i $j = 1, 2, \dots, m^*$, i disposant de n nombres reals y_i , elegir una funció $F: \mathfrak{R}^{m^*} \rightarrow \mathfrak{R}$ que satisfaci les condicions d'interpolació:

$$F(\check{x}_i) = y_i, \text{ amb } i = 1, 2, \dots, n. \quad (5.5)$$

La funció F està obligada a passar per tots els punts d'interpolació (\check{x}_i, y_i) .

Una vegada establerta la funció $F(\check{x}) = y$, si agafem punts de l'espai d'entrada diferents als inicials (punts d'interpolació), n'obtidrem les imatges en l'espai de sortida. Per tant, la funció F estarà generalitzant, és a dir, serà capaç de donar resposta a punts diferents dels d'interpolació.

La figura següent mostra el procés d'interpolació i de generalització de dades en un model lineal.

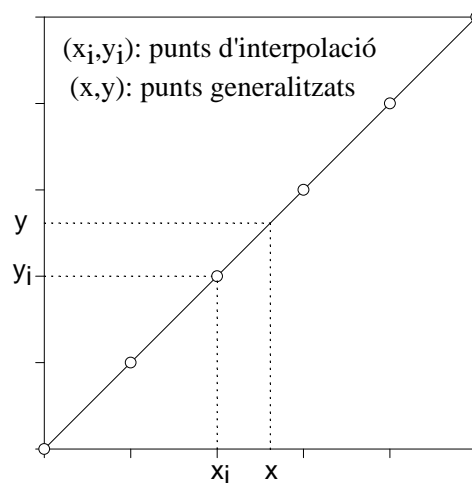


Figura 5.2.

Les parelles (x_i, y_i) són les dades d'interpolació que serveixen per determinar el model de regressió. A partir d'aquestes, el model permet generalitzar els resultats a parelles (x, y) que no formen part de les parelles d'interpolació (aquelles que calibren el model matemàtic).

5.1.2. Xarxes neuronals artificials

Com s'ha explicat anteriorment, les xarxes neuronals artificials poden ser interpretades com a models generalitzats de regressió i interpolació, en general no lineals, multivariables i no paramètrics. Són molt emprades en casos en què es necessita crear un model complex de regressió. A [Cyb89] i [Fun89] s'explica que si l'estructura de la xarxa neuronal és suficientment gran, és a dir, quan hi ha suficients neurones ocultes, qualsevol funció contínua pot ser aproximada, amb un grau d'aproximació arbitrari (triat per nosaltres), seleccionant adequadament el valor dels paràmetres de la xarxa neuronal.

Les xarxes neuronals artificials pretenen ser una imitació matemàtica senzilla del model biològic del cervell i estan formades per subelements o neurones, connectades entre si per formar la xarxa.

El cos de la neurona biològica té un gran nombre de branques anomenades *dendrites*, les quals reben l'estímul elèctric i el passen al cos de la neurona. Allí els senyals elèctrics s'acumulen i quan aquest valor assoleix un llindar, la neurona és excitada i emet un estímul elèctric a través de l'àxon, el qual es troba connectat a les dendrites de les neurones veïnes a través de la sinapsi (punt de contacte entre diferents neurones). Les sinapsis alteren la intensitat d'aquest senyal d'acord amb un cert pes.

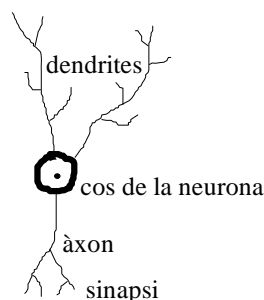


Figura 5.3.

En els models matemàtics de xarxes neuronals artificials es fa la hipòtesi següent: quan un senyal d'entrada x_i creua una sinapsi entre una neurona i i una neurona j , el seu valor queda multiplicat per un pes w_{ji} (vegeu la figura 5.4.).

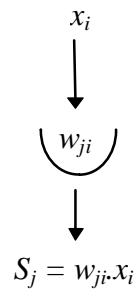


Figura 5.4.

Cada neurona té moltes dendrites (en la figura 5.5. en té a) i, per tant, rep molts senyals simultàniament. Cada neurona dóna un senyal de sortida y_j , que és generat a partir d'una transformació del senyal S_j d'entrada (en general, aquesta transformació no serà lineal).

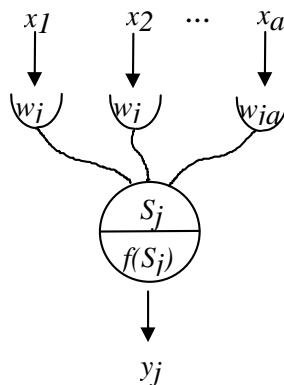


Figura 5.5.

El senyal provinent de les sinapsis o senyal d'entrada d'una neurona j és:

$$S_j = \sum_{i=1}^a w_{ji} \cdot x_i. \quad (5.6)$$

La sortida d'aquesta mateixa neurona és funció de S_j , complint-se:

$$y_j = f(S_j). \quad (5.7)$$

La forma de la funció $f(S_j)$ dependrà normalment del model de xarxa utilitzat.

Les neurones també s'anomenen *nodes* i, tal com s'ha explicat, normalment tenen diverses entrades i una única sortida. En les xarxes artificials les neurones s'agrupen per capes, de manera que, en general, les neurones de capes veïnes es connecten entre si, totes amb totes. Tota xarxa neuronal té una capa de nodes d'entrada (realment no compta com a capa, ja

que els seus nodes no fan cap operació matemàtica), una o diverses capes intermèdies o ocultes i una capa de sortida.

Pel que fa al nombre de neurones per capa, és molt senzill saber el nombre necessari de neurones per a les capes d'entrada i de sortida. Hi ha d'haver tants nodes d'entrada com variables d'entrada tingui el problema (m^*) i tantes neurones de sortida com classes d'objecte hi hagi definides (c). La problemàtica recau en la capa o les capes ocultes. Per saber-ne el nombre de neurones, no hi ha cap criteri exacte. En general, però, amb una sola capa oculta és suficient.

Suposem una xarxa neuronal formada només per la capa d'entrada i la capa de sortida. Aquest és el model més senzill que hi ha i va ser una de les primeres estructures estudiades. Aquest tipus de xarxa s'anomena *perceptró*. La figura 5.6. mostra dues representacions diferents d'aquest model: la primera correspon a una representació gràfica on les rodones són les neurones i els quadrats indiquen les entrades (la capa d'entrada no està formada per neurones, ja que en aquesta no es realitza cap operació aritmètica), i la segona és una representació matricial.

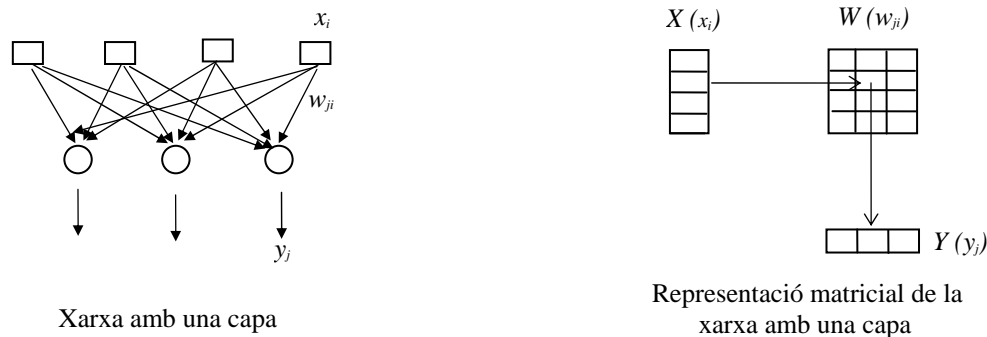


Figura 5.6.

La figura 5.7. mostra l'estructura típica d'una xarxa amb dues capes de neurones: una capa oculta més una capa de sortida. Es veu que les neurones de les diferents capes es troben connectades totes amb totes. Aquesta estructura s'anomena *feedforward* i és molt utilitzada.

Les característiques de les xarxes neuronals artificials depenen principalment de:

- Les operacions aritmètiques que realitzen les neurones.
- L'arquitectura de la xarxa, és a dir, el nombre de capes, les connexions entre les neurones, el tipus de funció de transferència de les neurones, etc.

- El procés d'aprenentatge (en l'àmbit de xarxes neuronals es parla d'*aprenentatge* en comptes de *calibratge*), en el qual s'adapten els valors dels pesos per arribar al resultat correcte.

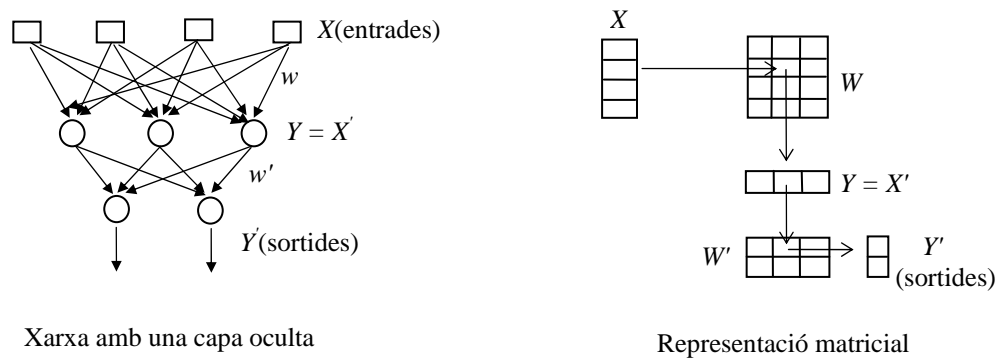


Figura 5.7.

Xarxes *backpropagation*

Les xarxes *backpropagation* representen el tipus de xarxa neuronal més utilitzat. Segueixen l'arquitectura *feedforward* explicada anteriorment (les neurones d'una capa tenen connexions amb totes les de les capes veïnes).

En aquest tipus de xarxes, la funció de transferència més àmpliament utilitzada en les neurones ocultes és de tipus sigmoïdal:

$$y_j = f(S_j) = \frac{1}{1 + e^{-(\alpha \cdot S_j + \theta)}} \quad (5.8)$$

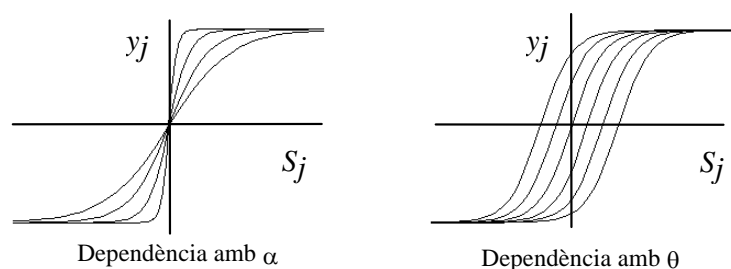


Figura 5.8.

Les figures anteriors mostren la dependència de la resposta de la neurona respecte dels paràmetres α i θ .

L'algorisme *backpropagation* va ser introduït el 1986 per Rumelhart, Hinton i Williams [Rum86] i representa un tipus especial de model d'aprenentatge (calibratge), fins ara el més aplicat. Vegeu [Lip87], [Kos92], [Wal92], [Bah93], [Gas93], [Son93], [Wyt93] i [Lie94].

Quan es parla d'*aprenentatge* (calibratge), es fa referència al procés mitjançant el qual l'algorisme va ajustant el valor dels pesos dels lligams entre les neurones fins a arribar a uns valors tals que minimitzen l'error, que es calcula com la diferència entre la sortida generada per la xarxa i la sortida coneguda. Aquest és un model d'aprenentatge supervisat, ja que constantment es té en compte la diferència entre la sortida instantània generada per la xarxa i la sortida coneguda.

L'algorisme *backpropagation* s'aplica a xarxes multicapa i consisteix en una propagació de les dades (valors de les variables) des de l'entrada cap a la sortida i en una altra propagació dels errors des de la sortida cap a l'entrada per tal d'anar corregint els valors dels pesos. Això es fa amb la finalitat de minimitzar l'error en la sortida. Les dades d'entrada passen a través de les diferents capes, de manera que la sortida de la capa 1 fa d'entrada a la capa 2 i així successivament. L'última sortida representa el conjunt de valors esperats; a partir d'aquesta es pot calcular l'error. L'objectiu és minimitzar l'error; per això aquest s'utilitza per corregir els pesos de la capa de sortida. Tot seguit es corregeixen els pesos de la penúltima capa i així successivament fins a arribar a l'entrada. L'error, per tant, és realimentat capa a capa des de la sortida fins a l'entrada.

L'algorisme *backpropagation* es pot resumir en els cinc punts següents:

Pas 1.- Cal inicialitzar tots els pesos amb valors aleatoris petits.

Pas 2.- S'entra el conjunt de valors del vector d'entrada i el conjunt de valors del de sortida (entrem un objecte de calibratge). Si fem servir la xarxa com a classificador, prendrem totes les sortides de valor 0 excepte la corresponent a la classe de l'objecte d'entrada, que serà 1.

Pas 3.- Es calculen de les sortides actuals y_1, \dots, y_c (pel temps t) a partir de la funció sigmoïdal.

Pas 4.- S'adapten els pesos fent servir un algorisme iteratiu. Es comença pels nodes de sortida i es va tirant enrere fins a arribar a la capa d'entrada. El procés d'ajust dels pesos és el següent:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \Rightarrow w_{ji}(t+1) = w_{ji}(t) + \eta \cdot \delta_j \cdot y_i.$$

η : taxa d'aprenentatge. Determina la velocitat amb la qual la xarxa aprèn. Permet controlar els canvis dels pesos durant el procés de calibratge. S'agafa entre 0,1 i 0,9.

δ_j : representa un terme d'error pel node j . Si $\delta_j = 0$, llavors $\Delta w_{ji} = 0$.

- Si j és un node de sortida: $\delta_j = y_j(1 - y_j)(d_j - y_j)$.

(d_j : valor correcte del node j ; y_j : valor actual del node j)

- Si j és un node ocult: $\delta_j = y_j(1 - y_j) \cdot \sum_{k=1}^n \delta_k \cdot w_{kj}$.

(k : cobreix tots els nodes de la capa següent on es troba el node j .)

Normalment la convergència és més ràpida introduint un terme α anomenat *momentum*. Aquest suavitza els canvis en els pesos:

$$w_{ji}(t+1) = w_{ji}(t) + \eta \cdot \delta_j \cdot y_i + \alpha \cdot [w_{ji}(t) - w_{ji}(t-1)].$$

Aquest terme dona una certa inèrcia al procés d'aprenentatge; fa que es recordi de la variació anterior dels pesos. Si α és petit, s'oblida ràpidament dels valors anteriors dels pesos. En general, s'agafen els paràmetres α i η de manera que el valor de la seva suma sigui proper a la unitat.

Pas 5.- Es torna al pas 2 mentre l'error no sigui inferior a un valor lliandar. L'error quadràtic mitjà es calcula com:

$$E = \frac{1}{n} \cdot \sum_{i=1}^n \left[\sum_{j=1}^c (d_{ij} - y_{ij})^2 \right],$$

on les d_{ij} representen les sortides correctes per a l'objecte i , les y_{ij} són les sortides estimades per la xarxa per a l'objecte i , c és el nombre de sortides i n és el nombre d'objectes de calibratge.

El procés d'aprenentatge es pot considerar com una recerca del mínim global d'una superfície d'error D -dimensional, on D és el nombre total de pesos de la xarxa. En xarxes multicapa de resposta no lineal, aquest mínim global ha d'existir forçosament, però també hi ha mínims secundaris. Quan la xarxa cau en un mínim secundari, es poden tenir problemes ja que hi ha un valor lliandar a partir del qual l'error no baixarà més. Per tant, cal dotar la xarxa d'algun mecanisme perquè surti d'aquest mínim secundari. El paràmetre α pot produir beneficis en la convergència de la xarxa, permet accelerar el procés d'aprenentatge en regions que tenen un gradient de la funció error constant i, a més, permet que la xarxa pugui sortir-se de la posició dels mínims secundaris, si els valors del paràmetre α són prou elevats.

Xarxes neuronals Radial Basis Functions (RBF neural networks)

Les Radial Basis Functions (RBF) són funcions matemàtiques de $\mathfrak{R}^n \rightarrow \mathfrak{R}$ que tradicionalment han estat emprades en problemes d'interpolació multidimensional. Hi ha definits molts tipus de RBF, però el més emprat de tots és, sens dubte, el de tipus gaussià:

$$RBF(\vec{x}) = \exp[-(\vec{x} - \vec{c})^2 / (2 \cdot \sigma^2)], \quad (5.9)$$

on \vec{x} i \vec{c} són vectors m^* dimensionals i representen, respectivament, la variable d'entrada i el centre de la RBF, mentre que el paràmetre σ modula l'amplada de la funció gaussiana.

Per a bibliografia sobre les xarxes neuronals Radial Basis Functions, consulteu [Che91], [Mat94] i [She96].

Les xarxes RBF estan formades per una capa d'entrada, una capa oculta i una capa de sortida. La capa d'entrada, com en la majoria de models de xarxes neuronals, té nodes que no són actius, ja que no efectuen cap operació matemàtica. Hi ha tants nodes d'entrada com variables tenen els objectes d'entrada. La capa oculta està formada per neurones, la funció de transferència de les quals és la RBF explicada en la fórmula 5.9. El nombre de neurones de la capa oculta és més difícil de conèixer *a priori*, tal com s'explica més endavant. La capa de sortida està formada per neurones, la funció de transferència de les quals és lineal. Hi ha tantes neurones de sortida com sortides hi hagi (en el nostre cas, classes d'imatges).

La figura 5.9. mostra la clàssica arquitectura d'aquest tipus de xarxes.

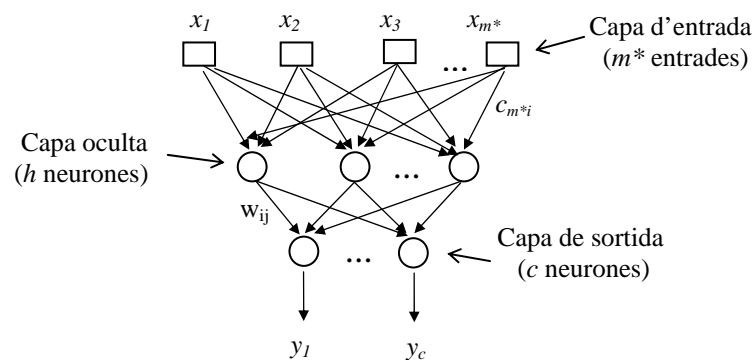


Figura 5.9.

En les figures 5.10. i 5.11. es mostren, respectivament, l'estructura i la funció de transferència típiques de les neurones de la capa oculta.

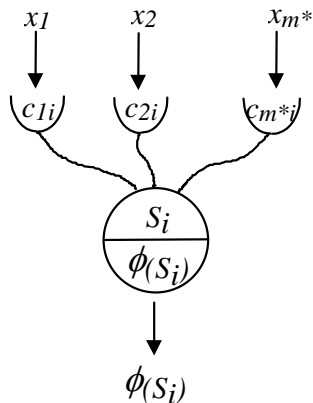


Figura 5.10.

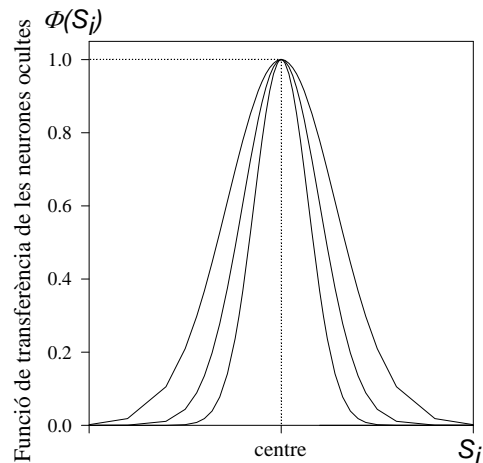


Figura 5.11.

L'entrada de cada neurona oculta ve donada per:

$$S_i = \sum_{r=1}^{m^*} (x_r - c_{ri})^2 = \|\mathbf{x} - \mathbf{c}_i\|^2. \quad (5.10)$$

Les neurones ocultes realitzen una transformació de tipus Radial Basis i les seves sortides vénen donades per:

$$\phi(S_i) = \phi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp[-(\|\mathbf{x} - \mathbf{c}_i\|)^2 / (2 \cdot \sigma^2)]. \quad (5.11)$$

El paràmetre σ normalment s'agafa igual per a totes les neurones ocultes i el seu valor s'ajusta durant la fase de calibratge.

Com es pot veure, les neurones ocultes donen una resposta profundament no lineal, el valor de la qual serà proper a la unitat si el vector de variables de l'objecte d'entrada és molt proper al centre de la neurona. La distància emprada és l'eucldiana.

Les neurones de la capa de sortida tenen una resposta lineal. Aquesta resposta és una combinació lineal de les sortides proporcionades per la capa oculta:

$$y_j = f(s_j),$$

$$\text{amb } s_j = \sum_{i=1}^h w_{ij} \cdot \phi(S_i) \quad \text{i} \quad f(s_j) = w_{oj} + \sum_{i=1}^h w_{ij} \cdot \phi(S_i). \quad (5.12)$$

La figura 5.12. mostra l'estructura d'una neurona de la capa de sortida:

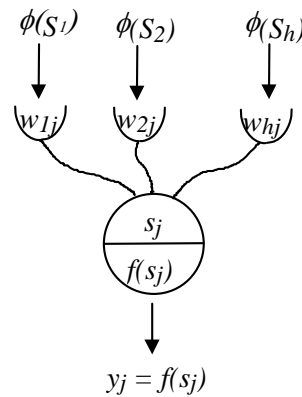


Figura 5.12.

La xarxa realitza una transformació $f: \mathcal{R}^{m^*} \rightarrow \mathcal{R}^c$, on el node j -èssim de la capa de sortida proporciona una funció de sortida f_j donada per l'expressió següent:

$$f_j(\mathcal{X}) = w_{oj} + \sum_{i=1}^h w_{ij} \cdot \phi(\|\mathcal{X} - \mathcal{C}_i\|). \quad (5.13)$$

Si es fa la transformació següent, les xarxes RBF poden ser vistes com a models de regressió lineal multivariable:

$$z_i = \phi(\|\mathcal{X} - \mathcal{C}_i\|). \quad (5.14)$$

Resulta que:

$$f_j(\mathcal{X}) = w_{oj} + \sum_{i=1}^h w_{ij} \cdot z_i. \quad (5.15)$$

La funció anterior és el típic model de regressió lineal múltiple, els paràmetres w_{ij} del qual es calculen pel mètode dels mínims quadrats.

Per tant, les xarxes neuronals RBF es poden veure com a tècniques matemàtiques amb una arquitectura típica d'una xarxa neuronal *feedforward* però amb una expressió matemàtica molt propera a la d'una regressió lineal múltiple.

Selecció del nombre de neurones ocultes en les xarxes RBF

Hi ha dos mètodes emprats per seleccionar el nombre de neurones de la capa oculta. Cada mètode porta a un mètode d'aprenentatge diferent de la xarxa neuronal.

- El primer consisteix a associar una neurona oculta a cada objecte de calibratge, de manera que el centre d'aquest serà el vector d'entrada corresponent a aquest objecte. Per tant, cada neurona oculta s'especialitza en un vector d'entrada determinat. Els avantatges d'aquest mètode són els següents: és molt simple, la selecció de centres és

única, utilitza màximament la informació dels objectes de calibratge i és molt desitjable quan el nombre de dimensions de les dades d'entrada és elevat. En canvi, presenta alguns inconvenients: l'elevat cost computacional, el problema d'*ill-condition* (problema que apareix quan s'inverteixen numèricament matrius grans. La matriu inversa que proporciona l'algorisme és aproximada i el seu resultat és molt sensible als valors dels coeficients) i el problema del sobreajustament (*over-fitting*) per part del model que genera la xarxa RBF (aquest problema apareix quan s'agafen més neurones del compte, ja que llavors el model aprèn tant els detalls individuals de cada objecte com els globals del conjunt d'objectes).

- El segon mètode de determinació del nombre de neurones de la capa oculta és força més complex, però també és més òptim, ja que permet minimitzar el nombre de neurones i determinar-ne alhora els centres. És un mètode més aconsellable i es basa en l'algorisme de mínims quadrats ortogonals que es troba detallat a [Che91]. Aquest algorisme consisteix a seleccionar un conjunt adequat de centres (vectors dels objectes de calibratge) entre un gran nombre de candidats. A cada pas de l'algorisme s'introdueix un nou centre (vector de les variables de cada objecte de calibratge) i, per tant, es maximitza l'increment de la variància explicada per la sortida desitjada. A més de minimitzar el nombre de neurones de la capa oculta, aquest algorisme permet evitar el problema d'*ill-condition*.

5.1.3. Mètodes de regressió

Regressió lineal múltiple (Multiple Linear Regression, MLR)

La MLR és una tècnica emprada en predicció que pretén relacionar linealment una matriu X de dades d'entrada conegudes amb una matriu Y de respostes. També s'anomena *Classical Least Squares* o *CLS*. Vegeu [Dra81], [Gel86], [Bee87], [Haa88], [San88] i [Kow91]. Aquest algorisme relaciona la matriu X amb la matriu Y mitjançant una matriu B de paràmetres desconeguts de la forma següent:

$$y_{ij} = \sum_{k=1}^m x_{ik} \cdot b_{kj} + \varepsilon_{ij}, \quad (5.16)$$

on:

$i = 1, 2, \dots, n$ (nombre d'objectes), $j = 1, 2, \dots, c$ (nombre de classes),

$k = 1, 2, \dots, m$ (nombre de característiques).

Per a cada objecte (índex i), el model ha de proporcionar tantes respostes com classes hi hagi (índex j).

L'expressió anterior es pot representar en forma matricial:

$$Y_{(n,c)} = X_{(n,m)} \cdot B_{(m,c)} + E_{(n,c)}, \quad (5.17)$$

on E és la matriu d'errors o desviacions de les respostes proporcionades pel model respecte a les respostes reals.

La MLR fa servir el mètode dels mínims quadrats per trobar les estimacions \bar{b}_{kj} (per això també s'anomena *Classical Least Squares* o *CLS*), de manera que la següent suma de quadrats F sigui mínima:

$$F = \sum_{i=1}^n \left(y_{ij} - \sum_{k=1}^m x_{ik} \cdot b_{kj} \right)^2. \quad (5.18)$$

[Dra81] demostra que la solució s'obté de l'equació:

$$X^T \cdot X \cdot B = X^T \cdot Y. \quad (5.19)$$

De l'equació anterior resulten les possibilitats següents:

- Si $m \cdot c > n$, tenim més incògnites que equacions. No és el que volem.
- Si $m \cdot c = n$, tenim igual nombre d'incògnites que d'equacions, cosa molt difícil en la pràctica.
- Si $m \cdot c < n$ i $\text{rang}(X) = m$ (té rang complet), $X^T \cdot X$ té inversa i la solució $B = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$ és única.
- Si $m \cdot c < n$ i $\text{rang}(X) < m \cdot c$, una solució és $B = (X^T \cdot X)^{\sim} \cdot X^T \cdot Y$, on el superíndex \sim indica la inversa generalitzada (vegeu l'annex).

El model resultant és:

$$Y_{(n,c)} = X_{(n,m)} \cdot B_{(m,c)}. \quad (5.20)$$

El problema que presenta aquest procediment és que només funciona si les m característiques utilitzades estan altament incorrelacionades, ja que si no les unes es poden posar com a combinacions lineals de les altres i el rang de la matriu X disminueix. Per tant, cal tenir molta cura a l'hora de triar les característiques. En general, aquest no serà un bon sistema per resoldre el nostre problema.

PCR (Principal Components Regression)

La PCR és una tècnica multivariant que, com la MLR, serveix per modelar la relació entre un conjunt de variables d'entrada X i un conjunt de variables de sortida Y . El model que realitzen les dues és del tipus:

$$Y_{(n,c)} = X_{(n,m)} \cdot B_{(m,c)} + E_{(n,c)}.$$

Vegeu [Dra81], [Gel86], [Bee87], [Lor87], [Haa88], [San88], [Mar89] i [Kow91]. A diferència de la MLR, no treballa directament amb les variables d'entrada X , sinó que opera amb els seus components principals. Es treballa amb el nombre de components principals que es cregui més convenient, la qual cosa permet eliminar el soroll i reduir les dimensions del nostre problema.

La PCR té dos passos clau. El primer es basa en el càlcul de la matriu V de vectors propis (matriu de *loadings*) de $X^T \cdot X$ i el còmput de la matriu de *scores*:

$$T = X \cdot V. \quad (5.21)$$

El segon pas consisteix en la regressió de Y respecte de la matriu T :

$$Y = T \cdot B = X \cdot V \cdot B = X \cdot \hat{B}. \quad (5.22)$$

L'algorisme utilitzat per fer el model PCR està detallat a continuació:

1.- Centrar o autoescalar la matriu $X_{(n,m)}$ columna per columna.

2.- Fer la descomposició en valors singulars de X :

$$X_{(n,m)} = U_{(n,m)} \cdot \Sigma_{(m,m)} \cdot V_{(m,m)}^T.$$

3.- Triar el nombre m^* convenient de components:

$$X_{(n,m)} = U_{(n,m^*)} \cdot \Sigma_{(m^*,m^*)} \cdot V_{(m^*,m)}^T.$$

4.- Calcular els coeficients de \hat{B} a partir de:

$$\hat{B}_{(m,c)} = V_{(m,m^*)} \cdot \Sigma_{(m^*,m^*)}^{-1} \cdot U_{(m^*,n)}^T \cdot Y_{(n,c)}.$$

5.- Fer la predicció d'objectes diferents a partir de:

$$Y'_{(n',c)} = X'_{(n',m)} \cdot \hat{B}_{(m,c)}.$$

LRR (Latent Root Regression)

Aquest és un mètode de regressió pràcticament idèntic a la PCR. Vegeu [Web74], [Mon82] i [San88]. La diferència fonamental radica en el fet de fusionar-se les matrius $X_{(n,m)}$ i $Y_{(n,c)}$ en una única matriu anomenada XY , tal com es mostra en la figura 5.13.

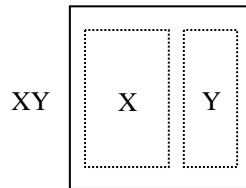


Figura 5.13.

Quan es té aquesta nova matriu, l'algorisme de càlcul del mètode és el següent:

1.- Centrar o autoescalar la matriu $XY_{(n,m+c)}$ columna per columna.

2.- Fer la descomposició de XY en valors singulars:

$$XY_{(n,m+c)} = U_{(n,m+c)} \Sigma_{(m+c,m+c)} V_{(m+c,m+c)}^T \quad .$$

3.- Triar el nombre m^* convenient de components i construir la matriu X_{UV} :

$$X_{UV(m^*,m^*)} = U_{(m^*,n)}^T X_{(n,m)} V_{(m,m^*)}$$

4.- Calcular els coeficients de \hat{B} a partir de:

$$\hat{B}_{(m,c)} = V_{(m,m^*)} X_{UV}^{-1} U_{(m^*,n)}^T Y_{(n,c)}$$

5.- Fer la predicció dels objectes del conjunt de test a partir de:

$$Y'_{(n',c)} = X'_{(n',m)} \hat{B}_{(m,c)}$$

PLS (Partial Least Squares)

És un dels millors mètodes de regressió. En general, funciona una mica millor que el PCR. Vegeu [Gel86], [Bee87], [Lor87], [Man87], [Haa88], [San88], [Mar89], [Kow91], [Lin93] i [Wak93]. És similar a la PCR ja que tant la matriu X de característiques com la matriu Y de respostes supervisades es descomponen en un producte de dues matrius: la matriu de *scores* i la matriu de *loadings*. Però mentre que durant l'etapa de calibratge la PCR no utilitza la informació de la matriu de respostes Y , la PLS sí que l'utilitza. Els *loadings* de X

es calculen a partir dels *scores* de Y i els *loadings* de Y es calculen a partir dels *scores* de X . També cal escollir el nombre òptim de variables m^* .

L'algorisme d'aquest mètode es detalla a continuació i s'ha de repetir per a totes les columnes de la matriu Y (quan es parla del vector y es fa referència a una columna de la matriu Y):

1.- $X_h = X$

Repetir des de $h = 1$ fins a $h = m^*$, augmentant h d'1 en 1

2.- $p_{h(m,1)} = X_h^T \cdot y / \|X_h^T \cdot y\|$, $t_{h(n,1)} = X_h \cdot p_{h(m,1)} / \|X_h^T \cdot y\|$.

3.- $X_h = \left[I_{(n,n)} - t_{h(n,1)} t_{h(n,1)}^T \right] X_h \cdot \left[I_{(m,m)} - p_{h(m,1)} p_{h(m,1)}^T \right]$.

4.- $P_{col,h} = p_h$, $T_{col,h} = t_h$.

5.- $Q = T^T \cdot X \cdot P$.

6.- $svd(Q) = U_Q \cdot S_Q \cdot V_Q^T$.

7.- Pseudoinversa de X : $\hat{X}^- = P \cdot V_Q \cdot S_Q^{-1} \cdot (T \cdot U_Q)^{-1}$.

8.- $b = \hat{X}^- \cdot y$.

9.- $B_{col,h} = b$.

Final de la repetició.

10.- $y' = X \cdot B_{l\text{ columna}}$

Agafant la h -èsima columna de B es tenen en compte les primeres h variables.

Regressió contínua

És un mètode de regressió molt versàtil i potent (vegeu [Kow91]), ja que mitjançant l'ús d'un exponent variable n es pot variar el tipus de regressió. Com que aquest coeficient n pot adoptar qualsevol valor real, es poden obtenir infinits tipus de regressions. En particular, per $n = 1$ s'obté la PLS i per $n = \infty$ s'obté la PCR. Però el millor mètode de regressió pot correspondre a un valor de n que no correspongui ni a la PLS ni a la PCR.

L'algorisme d'aquest procediment és molt similar al de la PLS. Està detallat a continuació:

$$1.- X_h = X$$

Repetir des de $h = 1$ fins a $h = m^*$, augmentant h d'1 en 1

$$2.- \text{svd}(X_h) = U_h \cdot S_h \cdot V_h^T.$$

$$3.- p_h = V_h \cdot S_h^n \cdot a / \|V_h \cdot S_h^n \cdot a\|, \quad t_h = U_h \cdot S_h^{2n} \cdot a / \|U_h \cdot S_h^{2n} \cdot a\| \quad \text{amb } a = U_h^T \cdot y.$$

$$4.- P_{col.h} = p_h, \quad T_{col.h} = t_h.$$

$$5.- Q = T^T \cdot X \cdot P.$$

$$6.- \text{svd}(Q) = U_Q \cdot S_Q \cdot V_Q^T.$$

$$7.- \text{Pseudoinversa de } X: \quad \hat{X}^- = P \cdot V_Q \cdot S_Q^{-1} \cdot (T \cdot U_Q)^{-1}.$$

$$8.- b = \hat{X}^- \cdot y.$$

$$9.- B_{col.h} = b.$$

Final de la repetició.

$$10.- y' = X \cdot B_{1 \text{ columna}}.$$

Agafant la h -èssima columna de B es tenen en compte les primers h variables.

5.2. Mètodes de classificació supervisats

Aquests mètodes serveixen per decidir si un objecte pertany o no a una classe d'objectes prèviament definida. Tal com es tracten en aquest treball, tornen un valor entre 1 i 0 depenent si l'objecte pertany o no a una classe determinada.

5.2.1. Anàlisi discriminant

L'anàlisi discriminant és un conjunt extens de tècniques estadístiques multivariants que permeten estudiar les diferències entre dues o més classes d'objectes mútuament exclusives a partir dels valors de les diferents variables utilitzades per caracteritzar els objectes. Es recomana la bibliografia següent: [Fuk72], [Dud73], [Lac75], [Kle80], [Cua81], [Fuk90], [Joh92], [McI92], [Krz93] i [Hub94].

S'ha emprat molt en camps com ara:

- La psicologia: per a tests de selecció de personal, de predicció de quins estudiants seran bons, etc.

- La política científica: per conèixer el comportament dels governants o pronosticar el vot dels ciutadans.
- La sociologia: per estudiar el comportament dels nens.
- La meteorologia: per pronosticar el temps.
- La biologia: per classificar plantes, animals, etc.

L'anàlisi discriminant pot ser emprada per a dos objectius bàsics:

- Quan volem interpretar el grau d'adequació de la classificació que hem realitzat. En aquest cas, es pretén mesurar el nivell de discriminació global de les variables i veure quines són les que tenen més poder discriminatori.
- Quan volem assignar els objectes d'estudi a una de les classes conegudes, la més propera a l'objecte en qüestió. Aquest és el procés de *classificació* pròpiament dit.

Per al conjunt d'objectes d'una classe qualsevol C es pot calcular el centre de gravetat \bar{x}_c de la classe (valor mitjà) i la matriu de variàncies-covariàncies S_c . La distància entre un vector $x \in \mathcal{R}^{m^*}$ (m^* és el nombre de variables) i el centre de gravetat \bar{x}_c de la classe C ve donada per la distància de Mahalanobis:

$$d(x, \bar{x}_c) = \sqrt{(x - \bar{x}_c)^T \cdot S_c^{-1} \cdot (x - \bar{x}_c)}. \quad (5.23)$$

Un sistema molt senzill de classificació podria consistir a assignar el vector x a la classe C de manera que la distància de Mahalanobis $d(x, \bar{x}_c)$ sigui la més petita de totes les classes existents. Assignem x a la classe C si es compleix:

$$d(x, \bar{x}_c) < d(x, \bar{x}_i) \quad \forall i = 1, 2, \dots, c. \quad (5.24)$$

Classificador bayesià (Maximum Likelihood Classification)

És un mètode probabilístic de classificació. El classificador bayesià suposa que les m^* variables $x = (x_1, x_2, \dots, x_{m^*})$ que descriuen els objectes segueixen una certa distribució de densitat de probabilitat $f(x)$. Consulteu [Cua81], [Fuk90], [Joh92] i [McI92].

El mètode més comú és suposar una distribució normal multivariant (m^* -dimensional) $f(x/C)$ per a cada classe. De fet, en la pràctica es treballa directament amb probabilitats, ja que $p(x/C)$ és proporcional a $f(x/C)$. L'expressió matemàtica de la funció densitat de probabilitat condicional que un objecte pertanyi a una classe C determinada, ve donada per la fórmula següent:

$$f(x/C) = \frac{1}{(2\pi)^{m^*/2} \sqrt{|S|}} \cdot \exp\left[-\frac{1}{2} \cdot (x - \bar{x})^T \cdot S^{-1} \cdot (x - \bar{x})\right], \quad (5.25)$$

on :

S : matriu estimada de variàncies-covariàncies de les variables de la classe C .

\bar{x} : vector estimat de mitjanes associades a les variables x de la classe C .

$|S|$: determinant de la matriu de variàncies-covariàncies de la classe C .

En el cas de tenir només dues variables, la funció donada per l'expressió (5.25) es pot representar tal com mostra la figura 5.14., però si el nombre de variables és superior a dos, ja no és representable.

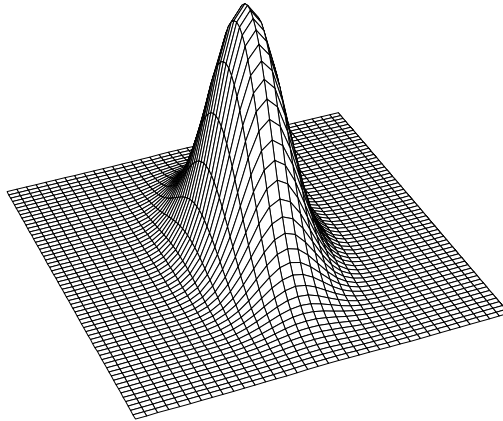


Figura 5.14.

La probabilitat que el vector de variables x d'un determinat objecte pertanyi a la classe C_i ve donada per l'expressió següent, aportada per Bayes:

$$p(C_i / x) = \frac{p(x / C_i) \cdot p(C_i)}{\sum_{j=1}^c p(x / C_j) \cdot p(C_j)} = \frac{p(x / C_i) \cdot p(C_i)}{p(x)}, \quad (5.26)$$

on:

$p(x/C_i)$: probabilitat que un vector de variables d'un objecte de la classe C_i adopti justament el valor x . El seu valor és proporcional al valor de $f(x/C_i)$ donat per l'expressió (5.25).

$p(C_i)$: probabilitat de la classe i , on $p(C_i) = n_i/n$, sent n_i el nombre d'objectes de cada classe i n el nombre total. En el cas d'igual nombre d'objectes per classe, es té $p(C_1) = \dots = p(C_i) = 1/c$.

$p(x)$: valor igual per a totes les classes ja que és una suma per a totes aquestes.

El problema és que $p(C_i/x)$ no es pot calcular directament i s'ha de trobar a partir de la fórmula (5.26). Acabem de veure que el denominador de (5.26) és el mateix per a totes les classes. A més, en el cas de tenir igual nombre d'objectes per cada classe, el càlcul de l'expressió (5.26) pot simplificar-se molt, ja que l'únic terme diferent per a cada classe és $p(x/C_i)$.

Això indica que l'objecte que s'ha de classificar es pot associar a la classe que presenti un valor de $p(x/C_i) \cdot p(C_i)$ més elevat i en la qual, a més, aquest valor sigui superior a un determinat llindar mínim.

Aquesta regla assigna l'objecte x a la classe C_i , que compleix:

$$p(x/C_i) \cdot p(C_i) > p(x/C_j) \cdot p(C_j), \quad \text{amb } i \neq j. \quad (5.27)$$

Com que $p(x/C_i)$ és proporcional a $f(x/C_i)$, resulta que:

$$f(x/C_i) \cdot p(C_i) > f(x/C_j) \cdot p(C_j). \quad (5.28)$$

Per tant, la **regla de màxima versemblança** (Maximum Likelihood Rule) assigna l'objecte x a la classe C_i si es compleix:

$$1.- B_{ij} = \frac{f(x/C_i)}{f(x/C_j)} > \frac{p(C_j)}{p(C_i)} = \frac{n_j}{n_i} \quad \forall i \neq j, \quad \text{amb } i, j \in 1, \dots, c. \quad (5.29)$$

$$2.- B_{ij}(x) > T \quad (\text{valor llindar o } \textit{threshold}).$$

(5.30)

Es fa necessari introduir un valor llindar T , perquè sense aquest tot objecte —encara que no pertanyés a cap de les classes definides— seria associat forçosament a una de les classes, la que donés un valor més alt de B_{ij} . Amb el valor llindar T , això ja no passarà.

LDA (Linear Discriminant Analysis)

Aquesta tècnica suposa que totes les classes d'objectes tenen igual matriu estimada de variàncies-covariàncies. Per tant, es té:

$$S_1 = S_2 = \dots = S_{m^*} = S. \quad (5.31)$$

A més, suposa les c classes C_i , amb $i = 1, 2, \dots, c$, normalment distribuïdes. Consulteu [Fuk90], [Joh92] i [McI92].

La funció densitat de probabilitat de la classe C_i s'estima aplicant la fórmula següent:

$$f(x / C_i) = \frac{1}{(2.\pi)^{m^*/2} \cdot \sqrt{|S|}} \cdot \exp\left[-\frac{1}{2} (x - \bar{x}_i)^T \cdot S^{-1} \cdot (x - \bar{x}_i)\right]. \quad (5.32)$$

La figura 5.15. mostra una representació en una dimensió de dues classes normalment distribuïdes amb iguals desviacions estàndards.

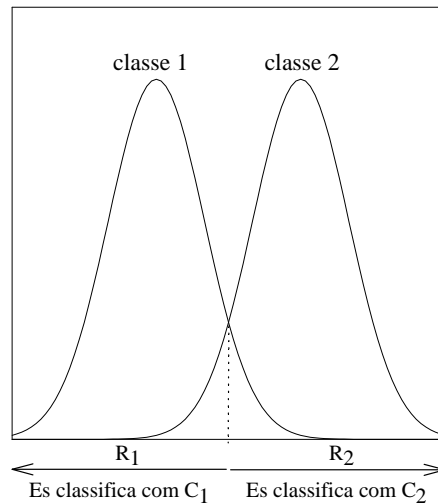


Figura 5.15.

Substituint en l'equació (5.29) l'expressió (5.32), resulta que:

$$\frac{f(x / C_i)}{f(x / C_j)} = \exp\left[(\bar{x}_i - \bar{x}_j)^T \cdot S^{-1} \cdot x - \frac{1}{2} \cdot (\bar{x}_i - \bar{x}_j)^T \cdot S^{-1} \cdot (\bar{x}_i + \bar{x}_j)\right] > \frac{p(C_j)}{p(C_i)}. \quad (5.33)$$

Aplicant logaritmes a l'expressió anterior resulta la funció discriminant lineal $L_{ij}(x)$:

$$L_{ij}(x) = \ln\left[\frac{f(x / C_i)}{f(x / C_j)}\right] = (\bar{x}_i - \bar{x}_j)^T \cdot S^{-1} \cdot \left[x - \frac{1}{2} \cdot (\bar{x}_i + \bar{x}_j)\right] \quad (5.34)$$

Es decideix que l'objecte de vector de variables x pertany a la classe C_i quan:

- 1.- $L_{ij}(x) > \ln\left[\frac{p(C_j)}{p(C_i)}\right] \quad \forall i \neq j, \text{ amb } i, j \in 1, 2, \dots, c.$
- 2.- $f(x/C_i) \geq T_{LDA,i}$ (valor llindar o *threshold* de la classe i -èssima). (5.35)

En aquest treball es proposa calcular el valor llindar de cada classe a partir de l'expressió següent:

$$T_{LDA,i} = [m_i + M_i]/2, \quad (5.36)$$

on m_i és el valor mínim que proporciona la gaussiana de la classe i -èssima a tots els objectes de calibratge pertanyents a aquesta classe i M_i és el valor màxim de la gaussiana de la classe i -èssima a tots els objectes de calibratge que no pertanyen a aquella.

El mètode que s'ha de seguir consisteix a calcular el vector d' m característiques discriminants de cada objecte, per així formar la matriu de característiques de cada una de les classes. Aplicant un mètode adequat de reducció de dimensions (vegeu el capítol 3) es pot passar d' m característiques a m^* variables. Llavors es poden determinar la funció de densitat de probabilitat de cada classe, donada per l'expressió (5.32), i les funcions discriminants entre classes. Quan es vol classificar un objecte determinat, només s'ha de calcular el valor numèric de la funció discriminant entre classes i associar aquest objecte a la classe que presenti el valor de $L_{ij}(x)$ més elevat, sempre que el seu valor superi el valor llindar $T_{LDA,i}$.

Com que el terme $(\bar{x}_i - \bar{x}_j)^T \cdot S^{-1}$ de l'expressió (5.34) és un vector fila de components constants, es pot fer l'associació següent:

$$b^T = (\bar{x}_i - \bar{x}_j)^T \cdot S^{-1}, \quad (5.37)$$

$$c = -\frac{1}{2} \cdot (\bar{x}_i - \bar{x}_j)^T \cdot S^{-1} \cdot (\bar{x}_i + \bar{x}_j). \quad (5.38)$$

Resulta que:

$$L_{ij}(x) = b^T \cdot x + c, \quad (5.39)$$

on

$$L_{ij}(x) = b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_{m^*} \cdot x_{m^*} + c. \quad (5.40)$$

Observant l'expressió anterior s'entén perquè la funció $L_{ij}(x)$ s'anomena *funció discriminant lineal*.

La probabilitat de classificar erròniament un objecte en la classe C_i , quan realment pertany a la classe C_j , ve donada per:

$$p(C_i / C_j) = \text{prob}(L_{ij}(x) > \ln \left[\frac{p(C_j)}{p(C_i)} \right]) / x \in \text{classe } C_j = \int_{R_i} f(x / C_j) dx, \quad (5.41)$$

on R_i és la zona de l'espai corresponent a la classe C_i .

QDA (Quadratic Discriminant Analysis)

Aquesta tècnica suposa que totes les classes tenen diferent matriu estimada de variàncies-covariàncies. Per tant, s'obté:

$$S_i \neq S_j \text{ per } i \neq j. \quad (5.42)$$

A més, suposa les c classes C_i (amb $i = 1, 2, \dots, c$), normalment distribuïdes. Consulteu [Fuk90], [Joh92] i [Mc192].

La funció densitat de probabilitat de la classe C_i s'estima aplicant la fórmula següent:

$$f(x / C_i) = \frac{1}{(2\pi)^{m^*/2} \cdot \sqrt{|S_i|}} \cdot \exp\left[-\frac{1}{2}(x - \bar{x}_i)^T \cdot S_i^{-1} \cdot (x - \bar{x}_i)\right]. \quad (5.43)$$

La figura 5.16. mostra una representació en una dimensió de dues classes normalment distribuïdes amb diferents desviacions estàndards.

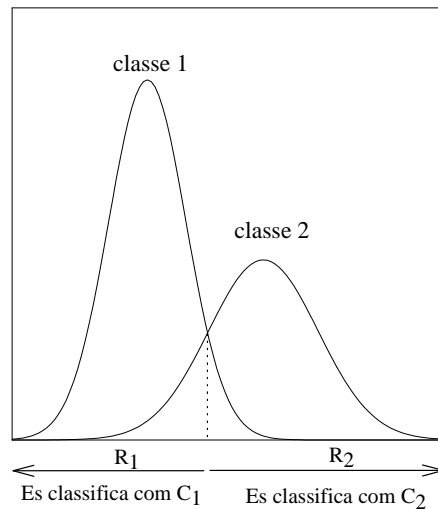


Figura 5.16.

Substituint en l'equació (5.29) l'expressió (5.43), resulta que:

$$\begin{aligned} \frac{f(x / C_i)}{f(x / C_j)} &= \\ &= \frac{\exp\left\{-\frac{1}{2}\left[x^T \cdot (S_i^{-1} - S_j^{-1}) \cdot x - 2 \cdot x^T \cdot (S_i^{-1} \cdot \bar{x}_i - S_j^{-1} \cdot \bar{x}_j) + \bar{x}_i^T \cdot S_i^{-1} \cdot \bar{x}_i - \bar{x}_j^T \cdot S_j^{-1} \cdot \bar{x}_j\right]\right\}}{|S_i|^{1/2} \cdot |S_j|^{-1/2}} > \frac{p(C_j)}{p(C_i)}. \end{aligned} \quad (5.44)$$

Aplicant logaritmes a l'expressió anterior resulta la funció discriminant quadràtica $Q_{ij}(x)$:

$$Q_{ij}(x) = \ln \left[\frac{f(x/C_i)}{f(x/C_j)} \right] =$$

$$= \frac{1}{2} \cdot \ln \frac{|S_j|}{|S_i|} - \frac{1}{2} \left[x^T \cdot (S_i^{-1} - S_j^{-1}) \cdot x - 2 \cdot x^T \cdot (S_i^{-1} \cdot \bar{x}_i - S_j^{-1} \cdot \bar{x}_j) + \bar{x}_i^T \cdot S_i^{-1} \cdot \bar{x}_i - \bar{x}_j^T \cdot S_j^{-1} \cdot \bar{x}_j \right]. \quad (5.45)$$

Es decideix que l'objecte de vector de variables x pertany a la classe C_i quan:

$$1.- Q_{ij}(x) > \ln \left[\frac{p(C_j)}{p(C_i)} \right] \quad \forall i \neq j, \text{ amb } i, j \in 1, \dots, c.$$

$$2.- f(x/C_i) \geq T_{QDA,i} \text{ (valor llindar o } \textit{threshold} \text{ de la classe } i\text{-èsima)}. \quad (5.46)$$

En aquest treball es proposa calcular el valor llindar de cada classe a partir de l'expressió següent:

$$T_{QDA,i} = \sqrt{m_i \cdot M_i}, \quad (5.47)$$

on m_i és el valor mínim que proporciona la gaussiana de la classe i -èsima a tots els objectes de calibratge pertanyents a aquesta classe i M_i és el valor màxim de la gaussiana de la classe i -èsima, a tots els objectes de calibratge que no pertanyen a aquella.

L'operativa d'aquest mètode és idèntica a la del LDA, però canviant $L_{ij}(x)$ per $Q_{ij}(x)$.

Si s'aplica la QDA en el cas de tenir classes amb dispersions idèntiques, s'obté la LDA. Quan les diferents classes tinguin dispersions diferents (és el cas més general), la QDA s'ajustarà millor a la realitat que la LDA. Quan les diferents classes tinguin dispersions similars, segurament els dos mètodes proporcionaran resultats molt similars.

5.2.2. Mètode SIMCA (Soft Independent Model of Class Analogy)

És un mètode de classificació que va ser desenvolupat per S. Wold [Wol76]. Va ser pensat per a situacions en les quals el nombre m^* de variables és gran en relació amb el nombre n d'objectes. Per a més informació, consulteu [Wol76], [Mal77], [Sjö79], [Drö87], [Fra89a], [Fra89b] i [McI92].

Un vegada s'ha calculat el vector m^* -dimensional de variables x de cada objecte, cada classe es representarà pel model de components principals. El nombre de components principals retenguts pot ser diferent per a cada classe (r_i és el nombre de components principals retenguts de la classe i -èsima, amb $i = 1, 2, \dots, c$). Cada objecte es classifica

d'acord amb de la distància relativa entre el seu vector de components principals i el model de la classe en qüestió.

Es defineixen:

- La matriu de variàncies-covariàncies de la classe i -èsima:

$$S_i = \frac{1}{n_i - 1} \cdot \sum_{k=1}^{n_i} (x_k - \bar{x}_i) \cdot (x_k - \bar{x}_i)^T, \text{ on } i = 1, \dots, c \text{ és l'índex de les classes.} \quad (5.48)$$

- Els vectors propis de la matriu simètrica S_i : $v_{i1}, v_{i2}, \dots, v_{im^*}$, que estan associats als valors propis $\lambda_{i1} \geq \lambda_{i2} \geq \dots \geq \lambda_{im^*}$. Com que la matriu S_i és simètrica, els seus vectors propis seran ortonormals i es complirà:

$$S_i = \sum_{k=1}^{m^*} \lambda_{ik} \cdot v_{ik} \cdot v_{ik}^T, \quad (5.49)$$

i, per tant:

$$S_i^{-1} = \sum_{k=1}^{m^*} v_{ik} \cdot v_{ik}^T / \lambda_{ik}. \quad (5.50)$$

- El nombre d'objectes de la i -èsima classe: n_i amb $n_1 + \dots + n_c = n$.
- El nombre de components principals retinguts de la classe i -èsima: r_i .
- La matriu estimada de variàncies-covariàncies de la classe i -èsima: V_i .

Aquest mètode divideix l'espai de components principals en dos subespais: el primari i el secundari. Aquest és un mètode de regularització, ja que els valors propis associats al subespai primari s'estimen de valor infinit i, per tant, es compleix:

$$1/\lambda_{ik} = 0, \text{ amb } k = 1, \dots, r_i. \quad (5.51)$$

En canvi, els valors propis del subespai secundari s'estimen com:

$$l_i = [(n_i - 1)(n_i - r_i - 1)] \cdot \sum_{k=r_i+1}^{m^*} \lambda_{ik}. \quad (5.52)$$

El mètode SIMCA assigna l'objecte amb vector de variables x a la classe que minimitza el quocient entre la posició de l'objecte i el valor mitjà de la suma dels quadrats dels residuals de cada classe:

$$\min_i \frac{d_i^2(x) / (m^* - r_i)}{\sum_{j=1}^n z_{ij} \cdot d_i^2(x_j) / [(m^* - r_i)(n_i - r_i - 1)]}, \text{ on:} \quad (5.53)$$

$$d_i^2(x) = \sum_{k=r_i+1}^{m^*} \left[(x - \bar{x}_i)^T \cdot v_{ik} \right]^2 \quad \text{i} \quad z_{ij} = (z_j)_i = \begin{cases} 1 & \text{si } x_j \in C_i \\ 0 & \text{si } x_j \notin C_i \end{cases}, \quad \text{amb } i = 1, 2, \dots, c.$$

El numerador de l'expressió (5.53) resulta:

$$d_i^2(x)/(m^* - r_i) = (x - \bar{x}_i)^T \cdot \sum_{k=r_i+1}^{m^*} v_{ik} \cdot v_{ik}^T \cdot (x - \bar{x}_i)/(m^* - r_i). \quad (5.54)$$

Frank i Friedman [Fra89a] han demostrat que el mètode SIMCA es pot interpretar com una regla de mínima distància que pot ser expressada de la forma:

$$\min_i \left[(x - \bar{x}_i)^T \cdot V_i^{-1} \cdot (x - \bar{x}_i) \right]^{1/2}.$$

L'expressió anterior es basa en l'estimació de la distància de Mahalanobis a la mitjana dels objectes de cada classe. Per tant, (5.53) es pot rescriure com:

$$\min_i \left[(x - \bar{x}_i)^T \cdot V_i^{-1} \cdot (x - \bar{x}_i) \right], \quad (5.55)$$

on:

$$V_i^{-1} = \frac{\sum_{k=r_i+1}^{m^*} v_{ik} \cdot v_{ik}^T}{[(n_i - 1)/(n_i - r_i - 1)] \cdot \sum_{k=r_i+1}^{m^*} \lambda_{ik}} \quad \text{és l'estimació de la matriu } S_i^{-1}. \quad (5.56)$$

La demostració passa per:

$$\sum_{j=1}^n z_{ij} \cdot d_i^2(x_j) = \sum_{j=1}^n \sum_{k=r_i+1}^{m^*} z_{ij} \cdot v_{ik}^T \cdot (x_j - \bar{x}_i) \cdot (x_j - \bar{x}_i)^T \cdot v_{ik}.$$

Però com que:

$$\sum_{j=1}^n z_{ij} \cdot v_{ik}^T \cdot (x_j - \bar{x}_i) \cdot (x_j - \bar{x}_i)^T \cdot v_{ik} = (n_i - 1) \cdot v_{ik}^T \cdot S_i \cdot v_{ik},$$

resulta que:

$$\sum_{j=1}^n z_{ij} \cdot d_i^2(x_j) = (n_i - 1) \cdot \sum_{k=r_i+1}^{m^*} v_{ik}^T \cdot S_i \cdot v_{ik} = (n_i - 1) \cdot \sum_{k=r_i+1}^{m^*} \lambda_{ik}. \quad (5.57)$$

Un dels mètodes utilitzats per avaluar el nombre de components principals que s'han de retenir per classe (r_i) és avaluar per a cada classe la funció indicadora IND (trobadempíricament), descrita per Malinowski [Mal77]. La funció IND assoleix un mínim quan

s'utilitza el nombre correcte r_i de components principals per classe. Aquesta funció es defineix com:

$$IND_i = \frac{\sqrt{\frac{\sum_{j=r_i+1}^{m^*} \lambda_j}{MM(NN - r_i)}}}{(NN - r_i)^2}, \quad (5.58)$$

amb $i = 1, \dots, c$ $r_i = 1, \dots, m^*$, MM : màxim(n, m^*) i NN : mínim(n, m^*).

El mètode SIMCA té dos defectes:

- 1er.- En agafar $1/\lambda_{ik} = 0$, en l'estimació de V_i^{-1} s'ignora tota la informació de les diferències entre classes aportada pel subespai primari.
- 2on.- En considerar només la distància de Mahalanobis entre un objecte x i la mitjana dels objectes de cada classe, s'ignora el terme de normalització $|V_i|^{-1/2}$ que apareix en la funció de densitat normal multivariant de la distribució de x en C_i (funció gaussiana multivariant). Aquest terme pot aportar una contribució important quan les matrius de variàncies-covariàncies de cada classe siguin diferents.

5.2.3. Mètode DASCO (Discriminant Analysis with Shrunken Covariances)

Frank i Friedman [Fra89a] van introduir unes modificacions en el mètode SIMCA per tal de superar alguns dels inconvenients que aquest presenta. Van anomenar el nou mètode DASCO.

L'opertiva del mètode DASCO és molt similar a la del mètode SIMCA. A l'igual que SIMCA, DASCO també estima la inversa de V_i fent una partició de l'espai m^* -dimensional de variables de cada classe C_i en dos subespais: el primari, la dimensió del qual és r_i , i el seu complementari o subespai secundari, de dimensió $m^* - r_i$. Els valors propis associats al subespai secundari s'estimen com els valors mitjans dels $m^* - r_i$ valors propis del subespai secundari:

$$\bar{\lambda}_i = \frac{\sum_{k=r_i+1}^{m^*} \lambda_{ik}}{(m^* - r_i)} \quad (5.59)$$

En contrast amb SIMCA, els valors propis associats al subespai primari no s'agafen infinits, sinó que s'agafen els r_i més grans de C_i . En aquest cas resulta:

$$V_i^{-1} = \sum_{k=1}^{r_i} v_{ik} \cdot v_{ik}^T / \lambda_{ik} + \sum_{k=r_i+1}^{m^*} v_{ik} \cdot v_{ik}^T / \bar{\lambda}_i. \quad (5.60)$$

El nombre de components principals r_i que s'han de retenir per classe es calcula de la mateixa manera que en el cas de SIMCA.

5.2.4. Mètode dels k veïns més propers (k Nearest Neighbours, k NN)

Aquest és un mètode de classificació no paramètric (significa que no suposa cap funció preestablerta de densitat de probabilitat). Aquest mètode classifica un objecte x assignant-li la classe més representada en els k objectes més propers, és a dir, es decideix la classe a la qual pertany examinant els k veïns més propers. Per a bibliografia sobre aquest mètode, consulteu [Cov67], [Wag71], [Fuk72], [Dev82], [McI92] i [Bun93].

L'avantatge principal d'aquest mètode és que té una estructura molt senzilla i, per tant, és molt fàcil de programar.

Cover i Hart [Cov67] ja apuntaven que un dels problemes d'aquest mètode és que no hi ha un nombre òptim k (amb $k \neq 1$). Per tant, ha de ser l'investigador el que ha de provar diferents valors de k i comparar els resultats de classificació. Cal tenir en compte que el temps de càlcul augmenta molt en fer-ho el valor de k . Per motius de temps de càlcul, diferents autors recomanen utilitzar valors de k compresos entre 3 i 5.

Un altre problema que presenta és que obliga a emmagatzemar tots els objectes de calibratge, perquè cada vegada que es vol classificar un objecte de test s'ha de calcular la distància entre aquest i tots els objectes de calibratge.

La figura 5.25. mostra els k objectes de calibratge més propers a l'objecte de test, representats en l'espai de les dues primeres variables:

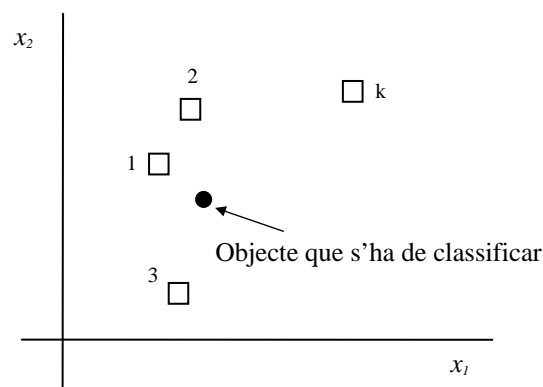


Figura 5.25

L'algorisme d'aquest mètode està explicat a continuació:

1. Representem els objectes en l'espai definit per les m^* variables, on cada objecte estarà determinat per unes coordenades.
2. Determinem els k veïns més propers a l'objecte que s'ha de classificar. Normalment s'utilitza la distància euclidiana, però també es pot utilitzar la distància de Mahalanobis (vegeu la fórmula 5.23) o altres. Els resultats poden ser diferents segons la definició de distància utilitzada.
3. Associem a la classe del veí més proper una puntuació k , a la del segon més proper $k-1$ i així successivament fins a 1.
4. El nostre objecte pertanyerà a la classe que presenti una puntuació més elevada.

Aquest mètode presenta la problemàtica que qualsevol objecte sempre tindrà una puntuació, pertanyi o no a alguna de les classes. Per tant, es pot establir una distància llindar. Si els objectes més propers es troben fora d'uns certs límits de seguretat, no se'ls atorga cap puntuació.

5.2.5. Paral·lelepípede que conté els objectes de calibratge

Aquest mètode és una proposta nova d'aquest treball i opera directament sobre les variables dels objectes del conjunt de calibratge. Per tant, requereix haver realitzat abans una reducció de dimensions (vegeu el capítol 3) en la matriu de característiques.

És una tècnica supervisada i no paramètrica (no pressuposa cap tipus de distribució de les dades del problema) que requereix un cost computacional molt baix.

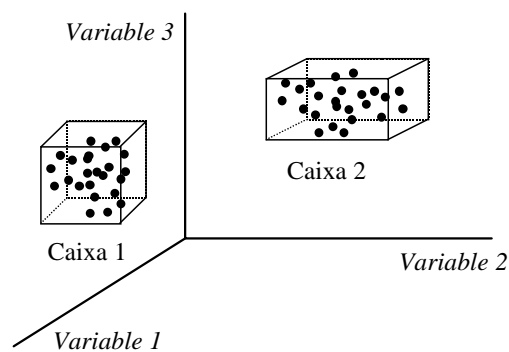


Figura 5.26.

L'operativa del mètode és la següent:

1. En la fase de calibratge cal trobar el paral·lelepípede o caixa de m^* -dimensions (sent m^* el nombre de variables) que contingui totes les coordenades dels objectes de calibratge d'una mateixa classe.
2. Es trobaran tantes caixes com classes d'objectes tingui el problema. Vegeu la figura 5.26.
3. En la fase de classificació, si les coordenades d'un objecte es troben dintre dels límits d'una caixa, s'assumirà que aquest objecte pertany a la classe d'aquesta caixa. En cas contrari, no hi pertanyerà.

En la fase de classificació d'objectes del conjunt de test és freqüent que succeeixi que, tot i que un objecte determinat pertanyi a una certa classe, les coordenades d'aquest es trobin properes a la caixa corresponent a aquesta classe, però fora dels límits d'aquesta. Per evitar que l'algorisme no reconegui l'objecte com a pertanyent a aquesta classe, el grau de pertinença d'aquest a la classe es calcularà a partir d'una funció que valgui 1 en l'interior de la caixa i decaigui cap a 0 a mesura que ens allunyem de la superfície exterior d'aquella (la caiguda a costat i costat de cada caixa es pot simular amb dues funcions exponencials decreixents, g_{1ij} i g_{2ij}). La figura 5.27. mostra una versió unidimensional d'aquest fet:

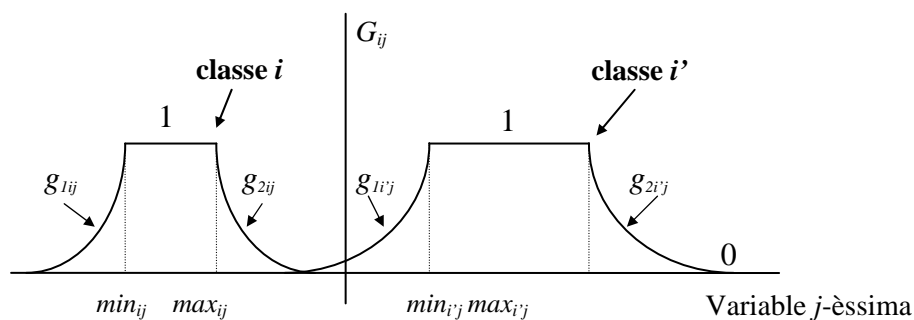


Figura 5.27.

En el cas de més dimensions s'opera de la mateixa manera que en el cas d'una dimensió, però es multipliquen les sortides G_{ij} degudes a cada dimensió. Les G_{ij} es defineixen com:

$$G_{ij} = \begin{cases} 1 & \text{si } x \in \text{límits de la caixa } i\text{-èssima} \\ g_{1ij} & \text{si } x < \min_{ij} \\ g_{2ij} & \text{si } x > \max_{ij} \end{cases} \quad (5.61)$$

$$g_{ij} = \exp\left(-k \cdot \left| \frac{x - \min_{ij}}{\max_{ij} - \min_{ij}} \right| \right) \quad \text{i} \quad g_{2ij} = \exp\left(-k \cdot \left| \frac{x - \max_{ij}}{\max_{ij} - \min_{ij}} \right| \right). \quad (5.62)$$

$i = 1, \dots, c$ (índex de classe), $j = 1, \dots, m^*$ (índex de dimensió o de variable), $k \in \mathcal{R}^+$.

El valor del paràmetre k modula la velocitat de decreixement de la funció exponencial i es pot ajustar durant el procés de calibratge. Per seleccionar el valor òptim de k s'assagen iterativament diferents valors i per cada un es calcula l'error global de classificació de tots els objectes de calibratge (es pot aplicar, per exemple, la validació creuada). El valor òptim de k serà el que produeixi l'error global de classificació més petit.

Per a un objecte d'entrada qualsevol, la sortida que proporciona l'algorisme per a la classe i -èsima es calcula com:

$$y_i = \prod_{j=1}^{m^*} G_{ij}, \quad \text{amb } i = 1, \dots, c \quad \text{i} \quad y_i \in [0, 1]. \quad (5.63)$$

5.2.6. Lògica difusa

La lògica difusa (Fuzzy Logic) va ser introduïda per primera vegada l'any 1965 per Zadeh [Zad65] i actualment s'està aplicant molt ([Kos92], [Jan95], [Rog95], [Jun98], [Mon98] i [Nav98]). Serveix per prendre decisions, però ofereix una resposta més real que la lògica clàssica, la qual només entén dos estats, o bé sí o bé no, mentre que la lògica difusa considera tot un ventall d'estats intermedis entre el sí i el no. Té un conjunt molt ampli d'aplicacions, que van des del reconeixement de formes, al camp dels sensors i el control de processos industrials ([Lee90], [Che92]).

Conjunts clàssics

Un conjunt clàssic es defineix de la manera següent:

$$A = \{ x / \text{condició} \}. \quad (5.64)$$

Per exemple, $A = \{ x / x > 6 \}$. No hi ha cap ambigüitat en els límits del conjunt. Pertanyen a aquest conjunt tots els objectes de valor més gran que 6 i no hi pertanyen tots els altres.

Conjunts difusos

Un conjunt difús es defineix de la manera següent:

$$A = \{ [x, \mu_A(x)] / x \in X \} \quad (5.65)$$

on $\mu_A(x)$ és el grau de pertinença o funció característica de x en el conjunt A i pot prendre valors entre 0 i 1, on X és una col·lecció d'objectes.

La definició d'un conjunt difús és que és l'extensió d'un conjunt clàssic on el grau de pertinença pot prendre valors continus entre 0 i 1. Si imposem que $\mu_A(x)$ valgui o bé 0 o bé 1 estarem definint un conjunt clàssic.

Propietats dels conjunts difusos

- Un conjunt difús A està inclòs en un conjunt B si: $A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$.
- La unió de dos conjunts A i B : $A \cup B = C$, amb $\mu_C(x) = \text{Màx}[\mu_A(x), \mu_B(x)]$.
- La intersecció de dos conjunts A i B : $A \cap B = C$, on $\mu_C(x) = \text{mín}[\mu_A(x), \mu_B(x)]$.
- El conjunt difús complementari de A és \bar{A} , on $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Tipus de funcions característiques

En els conjunts difusos es poden fer servir diferents tipus de funcions característiques. Les més habituals són les següents:

- Funció característica triangular (figura 5.17)
- Funció característica gaussiana (figura 5.18)
- Funció característica trapezoïdal (figura 5.19)
- Funció característica campana (figura 5.20)
- Funció característica sigmoïdal (figura 5.21)

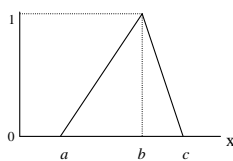


Figura 5.17.

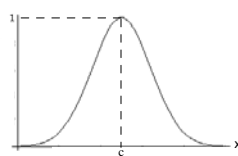


Figura 5.18.

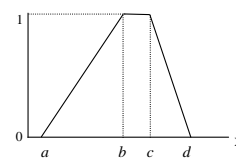


Figura 5.19.

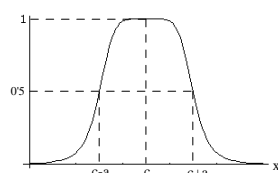


Figura 5.20.

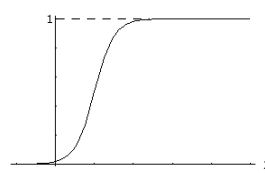


Figura 5.21.

Distribució dels conjunts difusos

Un mateix objecte pot pertànyer a diferents conjunts difusos al mateix temps, presentant un valor de la funció característica (entre 0 i 1) diferent per a cadascun. Per exemple, quan classifiquem objectes basant-nos en les seves característiques, podem dir que un objecte donat té un conjunt de característiques bo, regular o dolent en relació amb una certa classe d'objectes. A més, podem associar a cada nivell de classificació (bo, regular o dolent) un grau entre 0 i 1.

Anem a veure'n un exemple:

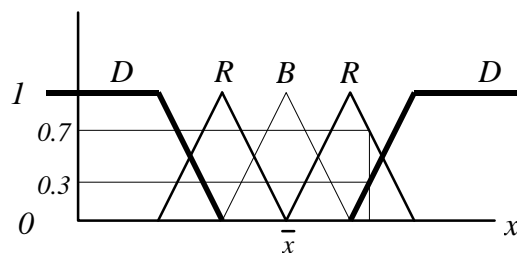


Figura 5.22.

En aquest cas, aquesta característica del nostre objecte presenta els nivells següents:

0 bo; 0,7 regular; 0,3 dolent.

Per simplificar les coses, a vegades es fa el canvi següent:

$$x' = |x - \bar{x}|, \quad (5.66)$$

on \bar{x} és el valor mitjà de la característica x . D'aquí resulta la característica següent:

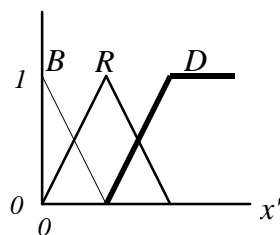


Figura 5.23.

Regles de decisió

Per decidir el valor de la sortida del sistema, és a dir, valorar si un objecte determinat presenta un conjunt de característiques similar al d'una certa classe, i, decidir per tant, si el

podem assimilar a aquesta, o, en canvi, si les seves característiques difereixen prou de les d'aquesta classe per decidir que no s'hi assembla, cal establir unes regles de decisió.

	$B(1,0)$	$R(0,5)$	$D(0,0)$
$B(1,0)$	$B(1,0)$	$R \hat{\uparrow}(0,75)$	$R(0,5)$
$R(0,5)$	$R \hat{\uparrow}(0,75)$	$R(0,5)$	$R \downarrow(0,25)$
$D(0,0)$	$R(0,5)$	$R \downarrow(0,25)$	$D(0,0)$

Taula 5.1

La taula de decisió anterior suposa que es tenen dues característiques per objecte i que classifiquem cadascuna d'aquestes amb un cert grau com a bona (B), regular (R) o dolenta (D).

Defuzzyficació

La *defuzzyficació* és el procés mitjançant el qual s'obté un valor numèric de la sortida (entre 0 i 1) a partir del conjunt de característiques de l'objecte en qüestió, de la distribució de conjunts difusos adoptada i de les regles de decisió aplicades.

Hi ha moltes formes de fer la *defuzzyficació*, però la més senzilla i la que més bons resultats acostuma a donar està explicada en l'exemple següent:

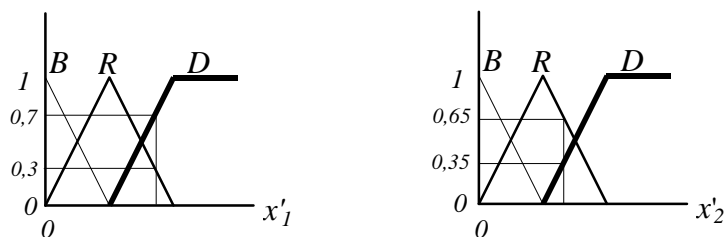


Figura 5.24.

Graus de pertinença de les característiques x_1 i x_2 als conjunts bo, regular i dolent:

$$1.- \mu_B(1) = 0, \quad \mu_R(1) = 0,3, \quad \mu_D(1) = 0,7$$

$$2.- \mu_B(2) = 0, \quad \mu_R(2) = 0,65, \quad \mu_D(2) = 0,35$$

$\mu_i(1)$	$\mu_i(2)$	$\min[\mu_i(1), \mu_i(2)]$	<i>Pes</i>
$\mu_B(1) = 0$	$\mu_B(2) = 0$	0	1,00
$\mu_B(1) = 0$	$\mu_R(2) = 0,65$	0	0,75
$\mu_B(1) = 0$	$\mu_D(2) = 0,35$	0	0,50
$\mu_R(1) = 0,3$	$\mu_B(2) = 0$	0	0,75
$\mu_R(1) = 0,3$	$\mu_R(2) = 0,65$	0,3	0,50
$\mu_R(1) = 0,3$	$\mu_D(2) = 0,35$	0,3	0,25
$\mu_D(1) = 0,7$	$\mu_B(2) = 0$	0	0,50
$\mu_D(1) = 0,7$	$\mu_R(2) = 0,65$	0,65	0,25
$\mu_D(1) = 0,7$	$\mu_D(2) = 0,35$	0,35	0,00

Taula 5.2.

Per tal de calcular la sortida total, cal fer totes les combinacions possibles entre els graus de pertinença de les dues característiques, tenint en compte les regles de decisió. Les combinacions possibles es veuen en la taula 5.2.

La sortida total és la mitjana aritmètica ponderada de tots aquests valors mínims multiplicats pel pes resultant de les regles de decisió:

$$S_{total} = \frac{1,00 \cdot 0 + 0,75 \cdot (0 + 0) + 0,50 \cdot (0 + 0,3 + 0) + 0,25 \cdot (0,3 + 0,65) + 0,00 \cdot 0,35}{0,3 + 0,3 + 0,65 + 0,35},$$

resultant que:

$$S_{total} = 0,2422.$$

5.3. Selecció del millor model de classificació

En aquest apartat s'expliquen els mètodes utilitzats per mesurar quin dels mètodes de predicció o de classificació és el que proporciona millors resultats. Per mesurar la bondat del model matemàtic de classificació cal fixar-se en els resultats de classificació d'objectes del conjunt de test. S'ha de tenir present que de la manera en què s'ha treballat, els mètodes de predicció i de classificació generen, per cada objecte d'entrada, tantes sortides com classes tingui definides el problema. Per tant, si es classifiquen n' objectes i el problema té definides c classes, en total es tindran $n' \times c$ sortides.

El percentatge d'errors de classificació es pot calcular a partir del nombre d'objectes classificats incorrectament o a partir del nombre de sortides errades. Com que els algorismes de classificació generen c sortides per cada objecte, quan un objecte no es classifica correctament pot ser que s'hagin errat d'1 a c sortides. És per aquest motiu que s'ha cregut més convenient treballar amb el percentatge de sortides errades que no pas amb el d'objectes classificats incorrectament.

Per avaluar la precisió del model matemàtic de classificació, en aquest treball s'utilitzaran simultàniament dues tècniques diferents, descrites a continuació.

Taxa de sortides errades

En els problemes de classificació és molt freqüent utilitzar la taxa de sortides errades com a mesura de l'exactitud del model matemàtic. Aquesta taxa es calcula com el tant per cent de sortides errades respecte a les sortides totals:

$$\% \text{ errors} = \frac{\text{sortides errades}}{\text{sortides totals}} \times 100. \quad (5.67)$$

L'expressió anterior s'ha de calcular sobre el total del conjunt d'objectes de test. Com més objectes de test hi hagi, més fiable serà el seu valor.

Molts autors també parlen dels errors tipus I i tipus II, definits a continuació:

Error de tipus I (ε_I): es produeix un error de tipus I quan un objecte es classifica com a no pertanyent a una classe quan realment hi pertany. Es calcula com:

$$\varepsilon_I = \frac{\text{nombre de sortides bones refusades}}{\text{nombre total teòric de sortides bones}}. \quad (5.68)$$

Error de tipus II (ε_{II}): es produeix un error de tipus II quan un objecte es classifica com a pertanyent a una classe quan realment no hi pertany. Es calcula com:

$$\varepsilon_{II} = \frac{\text{nombre de sortides dolentes acceptades}}{\text{nombre total teòric de sortides dolentes}}. \quad (5.69)$$

Com a mesura de la taxa d'encerts es dóna l'error mitjà de classificació, que és la mitjana ponderada dels errors de tipus I i de tipus II. Es calcula com:

$$\varepsilon = \varepsilon_I P(\text{sortides bones}) + \varepsilon_{II} P(\text{sortides dolentes}). \quad (5.70)$$

Si multipliquem l'error mitjà de classificació per 100, aquest coincideix amb el tant per cent d'errors.

En aquest cas, el millor classificador serà l'algorisme que produeixi el percentatge d'errors de classificació més baix.

Anem a veure en un exemple com s'han d'aplicar aquests criteris:

Tenim un problema amb $c = 3$ classes i $n' = 9$ objectes de test. La taula següent mostra en la part esquerra les sortides ideals de cada objecte i en la part dreta, les sortides generades per un determinat algorisme de classificació (en negreta hi ha les sortides errades).

Sortides conegudes			Sortides per un algorisme de classificació		
1	0	0	1	0	0
1	0	0	1	1	0
1	0	0	1	0	0
0	1	0	0	1	0
0	1	0	0	0	1
0	1	0	0	1	0
0	0	1	0	0	1
0	0	1	1	0	1
0	0	1	0	0	1

Taula 5.3

Els errors ε_I , ε_{II} i ε es calculen com:

$$\varepsilon_I = \frac{1}{9}, \quad \varepsilon_{II} = \frac{3}{18}, \quad \text{d'on resulta que:}$$

$$\varepsilon = \frac{1}{9} \cdot \frac{9}{27} + \frac{3}{18} \cdot \frac{18}{27} = \frac{4}{27}.$$

Això és el mateix que:

$$\% \text{ errors} = \frac{\text{sortides errades}}{\text{sortides totals}} \times 100 = \frac{4}{27} \times 100.$$

Per tant, s'acaba de comprovar en un exemple que els dos criteris són coincidents.

Mesura de l'exactitud de les sortides dels algorismes de classificació

El criteri de la taxa de sortides errades, encara que sigui capaç de caracteritzar molt bé l'eficiència del classificador utilitzat, no proporciona tota la informació possible sobre l'exactitud de les sortides. Pot succeir que dos algorismes de classificació, en classificar uns determinats objectes, generin la mateixa taxa de sortides errades, però que mentre que un d'aquests produeixi sortides molt properes a 0 o a 1 (quan l'objecte no pertanyi a la classe o quan hi pertanyi, respectivament), l'altre les generi properes a 0,5. Això significa

que el primer classificador identifica els objectes sense problemes, però el segon està al límit de la confusió. Per tant, es necessita més informació que la proporcionada pel criteri de la taxa de sortides errades; com a suport a aquesta informació es proposa utilitzar una mesura de l'exactitud de les sortides.

Com a mesura de l'exactitud de les sortides del model matemàtic de classificació es pot fer servir el PRESS (error quadràtic o Predictive Residual Error Sum of Squares) o el RRMSE (error relatiu quadràtic mitjà o Relative Root Mean Square Error), que es defineixen de la manera següent:

$$\text{PRESS} = \sum_{i=1}^n \sum_{j=1}^c (y_{ij} - y'_{ij})^2, \quad (5.71)$$

$$\text{RRMSE} = \frac{\sqrt{\sum_{i=1}^n \sum_{j=1}^c (y_{ij} - y'_{ij})^2}}{\frac{n}{|\bar{y}|}} \times 100, \quad (5.72)$$

on y_{ij} és la sortida teòrica (1 o 0) del conjunt d'objectes de test i y'_{ij} és la sortida proporcionada pel model matemàtic.

Com que en aquest treball les sortides dels classificadors es troben normalitzades dintre l'interval [0,1], s'utilitzarà el PRESS.

El millor classificador serà l'algorisme que, a més de classificar els objectes del conjunt de test amb una taxa de sortides errades més baixa, produeixi al mateix temps el PRESS més baix.