RFIDEC
Reconnaissance des formes et décision

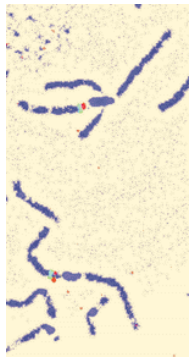
Master IAD – 1ère année - Université Paris 6
P. Gallinari
patrick.gallinari@lip6.fr
http://www-connex.lip6.fr/~gallinar/
Année 2009-2010

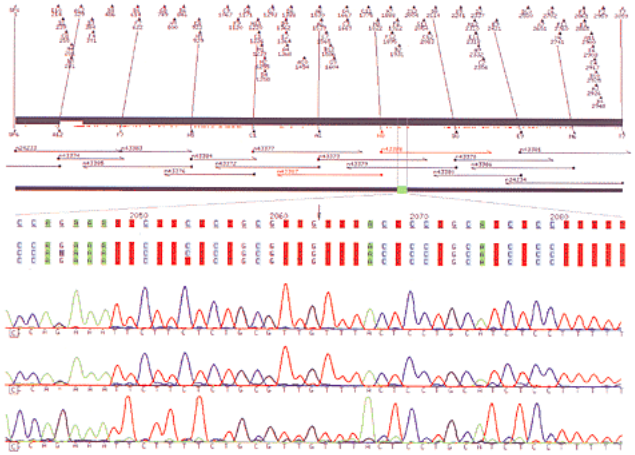
Reconnaissance des formes

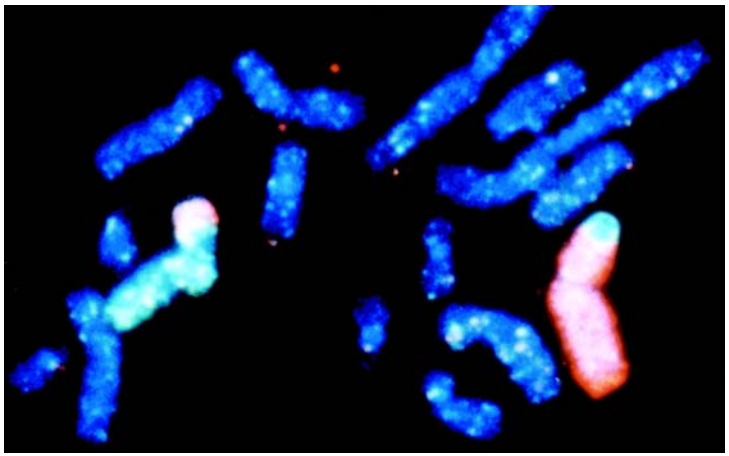
Reconnaissance des formes
Besoins

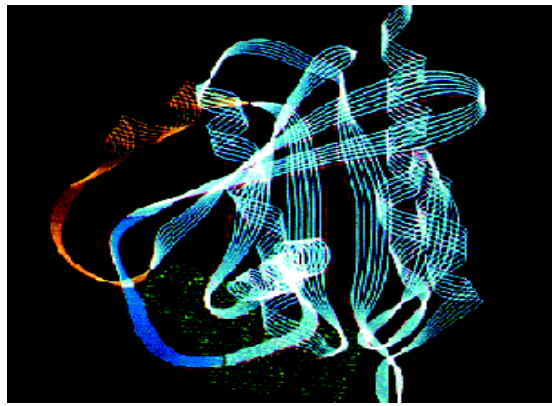
- Modéliser les données
- Prendre des décisions
- Prévoir
- Trouver des similarités entre données
- Visualiser des données

Biologie

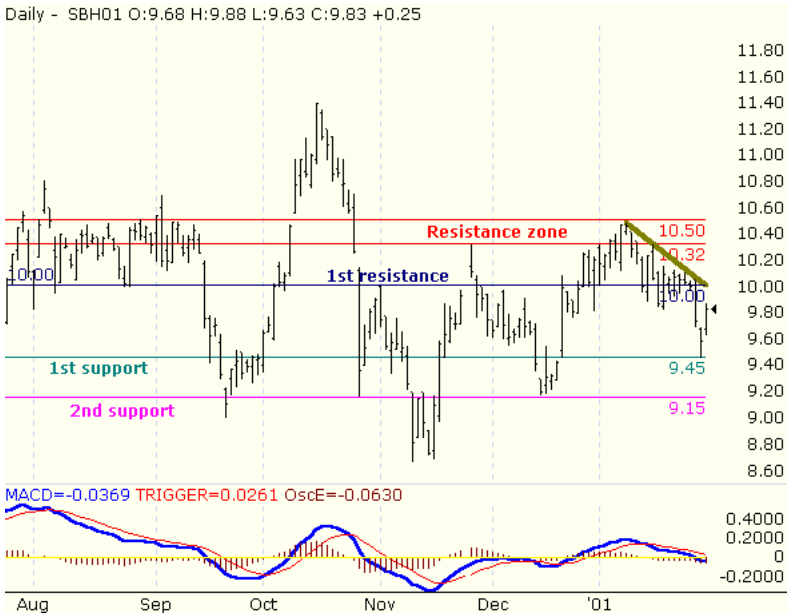


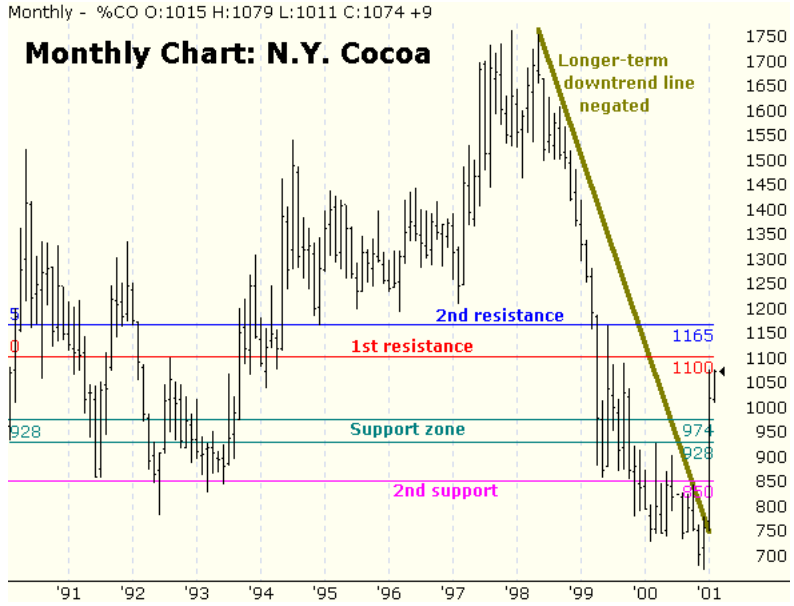




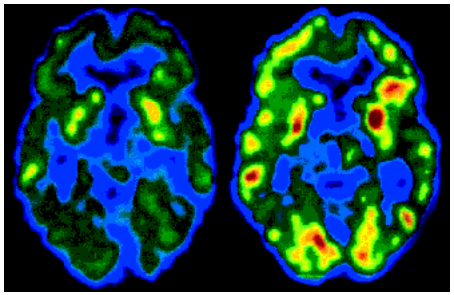


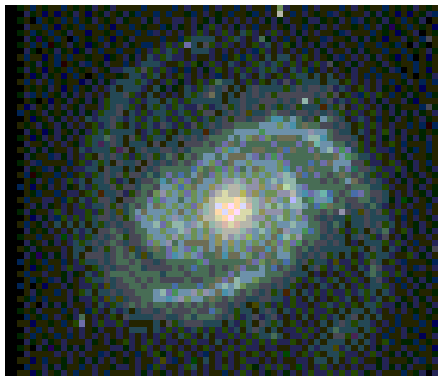
Séries boursières

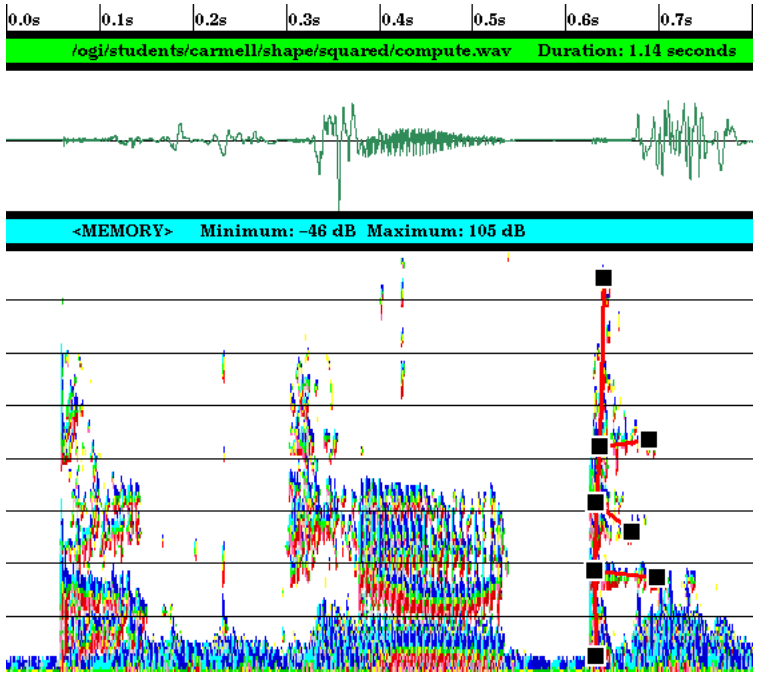




Images et Sons







Cadre

- Statistique
  - Phénomène régit par des lois inconnues dont on observe des réalisations
  - A partir de ces observations on veut inférer des connaissances, conclusions, etc
- Données
  - Représentent la principale connaissance sur le phénomène étudié
- 2 problématiques principales
  - Supervisé $D = \{x^i, d^i\}_{i=1..N}$
  - Non supervisé $D = \{x^i\}_{i=1..N}$

Classification

Décision Bayesienne

Fonctions discriminantes

Classification

Décision Bayesienne

Fonctions discriminantes

Théorie de la décision Bayesienne

- Cadre de la décision statistique
- Problème
  - Discrimination : décider de la classe d d'une observation x
- Objectif
  - Minimiser
    - Probabilité de décision erronée
    - Moyenne des coûts produits par des décisions erronées
- Hyp : on connaît toutes les probabilités

Notations

- x : donnée d'entrée
- Classes : C_1, .., C_p
- P_i : probabilité a priori de la classe C_i
- $P(x/C_i)$: probabilité conditionnelle de x pour la classe C
- $P(C_i/x)$: probabilité a posteriori de la classe
- Règle de décision

$$r : X \rightarrow C$$
$$x \rightarrow C_i$$

Décision Bayesienne : erreur minimale

- Affecter x à la classe la plus probable

$$r(x) = \arg\max_{C_i} \{P(C_i / x)\}$$

- Cette règle minimise la probabilité d'erreur (on regarde le cas à 2 classes)
  - Pour un x donné la probabilité d'erreur est

$$p(erreur|x) = \begin{cases} P(C_1|x) \text{ si on décide } C_2 \\ P(C_2|x) \text{ si on décide } C_1 \end{cases}$$

    - Qui est clairement minimisée par la règle de décision
  - En moyenne

$$p(erreur) = \int p(erreur, x)dx = \int p(erreur|x)\, p(x)dx$$

    - Si on prend la décision optimale pour tout x, ce sera vrai en moyenne

- Remarque
  - Avec cette règle de décision, la probabilité d'erreur est (cas 2 classes)

$$p(erreur|x) = \min(p(C_1|x), p(C_2|x))$$

- Autres formes de la règle de décision
  - Règle de Bayes

$$p(C_i / x) = \frac{P(x / C_i)p(C_i)}{p(x)} = \frac{P(x / C_i)p(C_i)}{\sum_j p(x / C_j)p(C_j)}$$

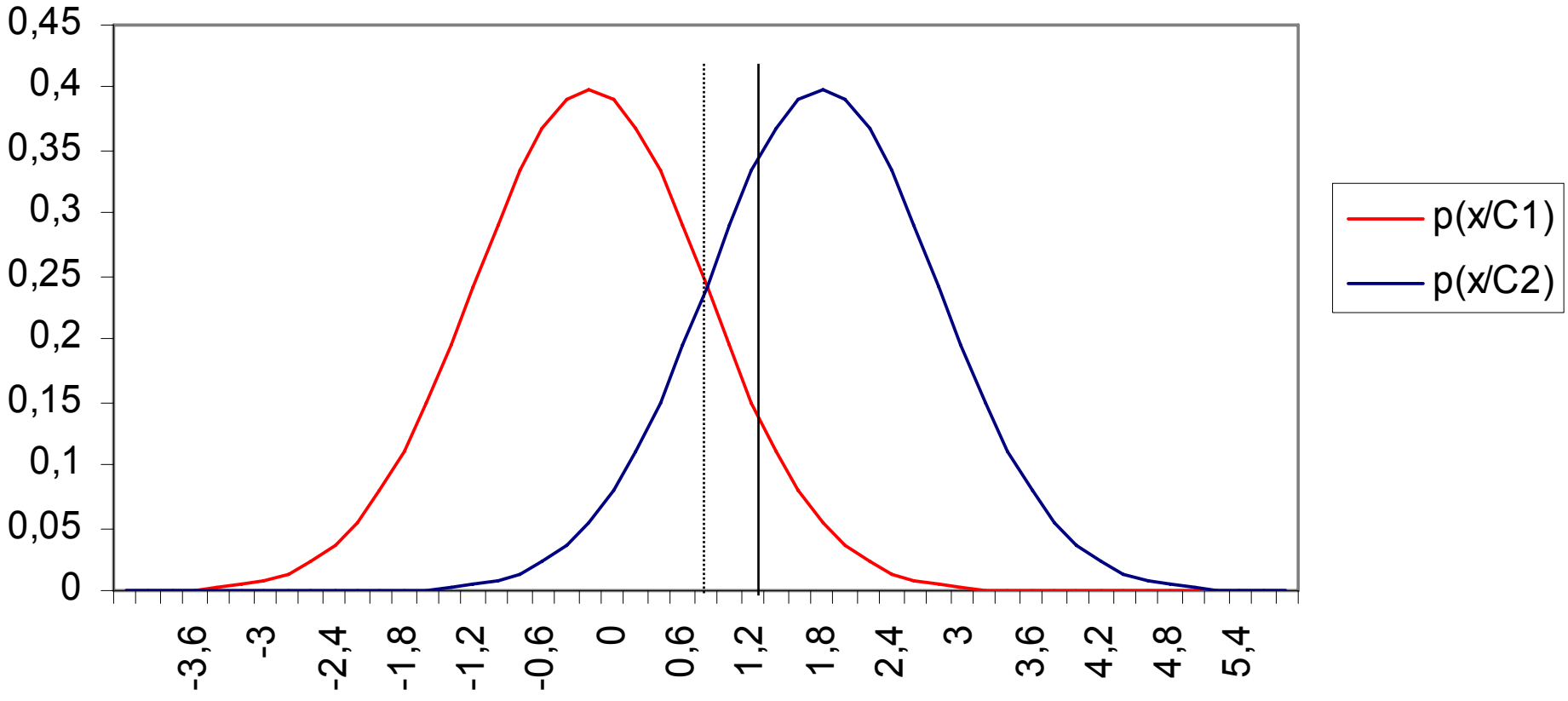
  - Décision

$$r(x) = \arg\max_{C_i}\{p(x / C_i)P(C_i)\}$$

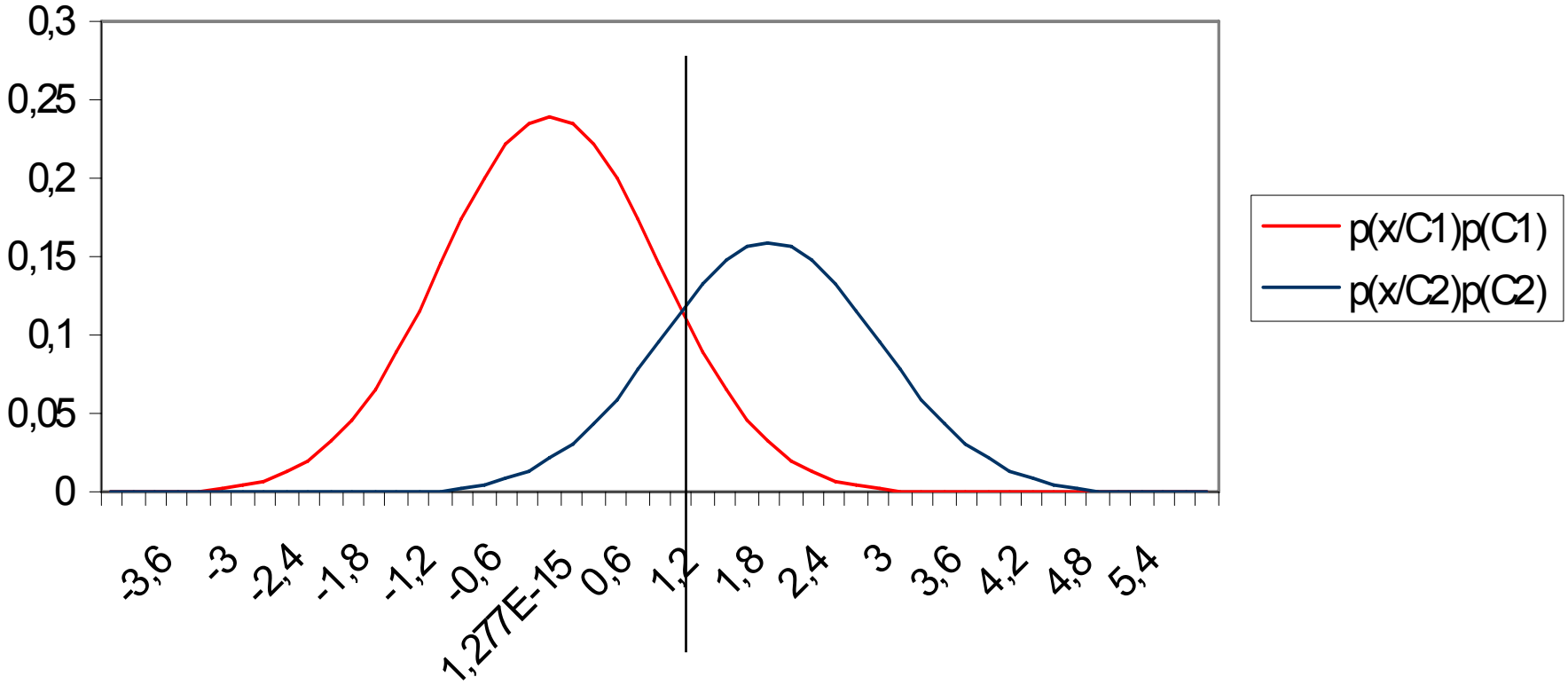
  - Cas à 2 classes

$$x \in C_1 \Leftrightarrow \frac{p(x / C_1)}{p(x / C_2)} > \frac{P(C_2)}{P(C_1)}$$

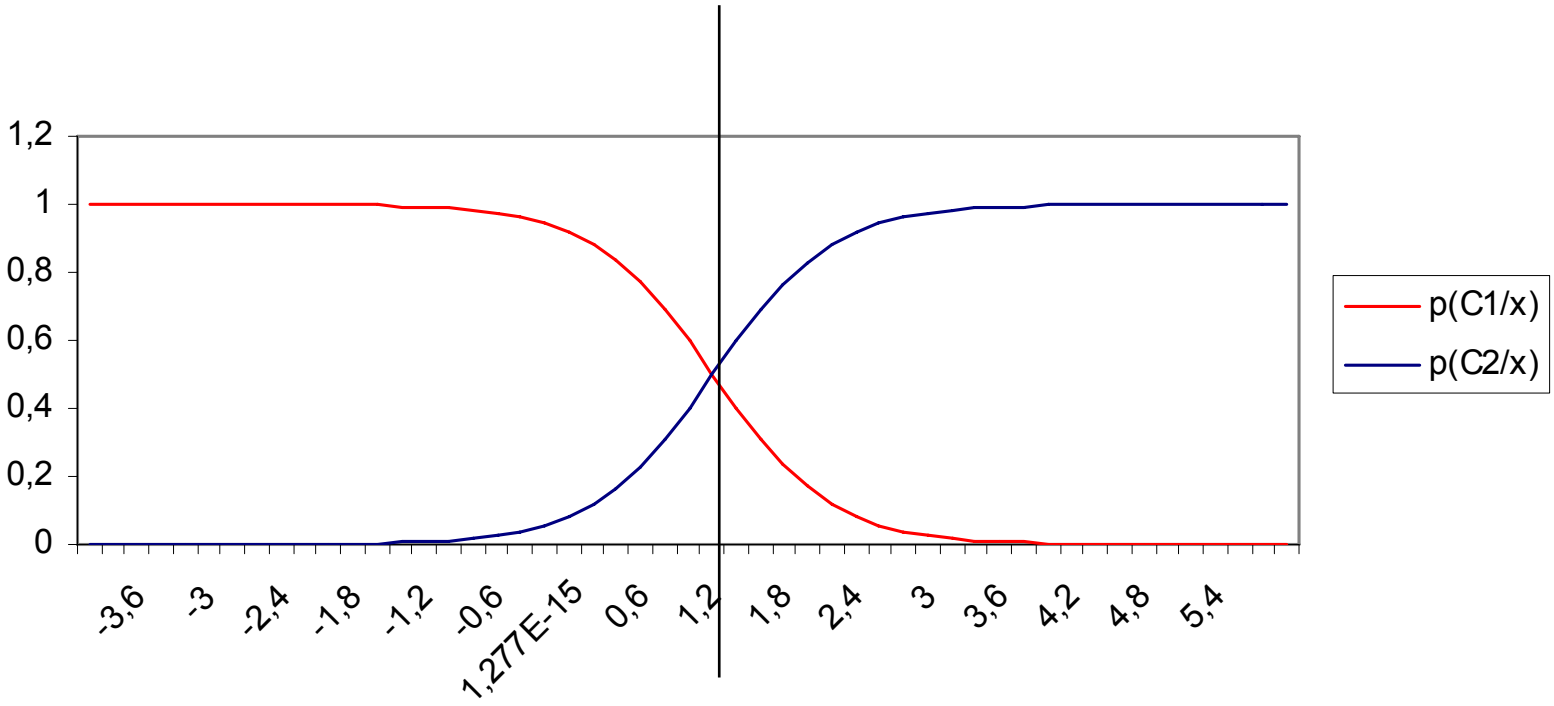
Densités


0,45
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
-3,6
-3
-2,4
-1,8
-1,2
-0,6
0
0,6
1,2
1,8
2,4
3
3,6
4,2
4,8
5,4
p(x/C1)
p(x/C2)

Densités pondérées



Probabilités a posteriori



Décision Bayesienne avec coût

- Toutes les décisions n'ont pas le même coût
- Exemple : diagnostic
  - Non détection : décider que le patient est sain (C_2) si il est malade (C_1)
  - Fausse alarme : décider qu'il est malade quand il est sain

- On introduit des coûts sur les décisions
  - C_{ij} : coût de la décision C_i quand la réalité est C_j
  - Coût associé à la décision C_i :

$$rc_i(x) = \sum_{j=1}^{p} C_{ij} P(C_j / x)$$

    - C'est le risque conditionnel a priori – espérance du coût du choix I
- Choix de coût minimum :

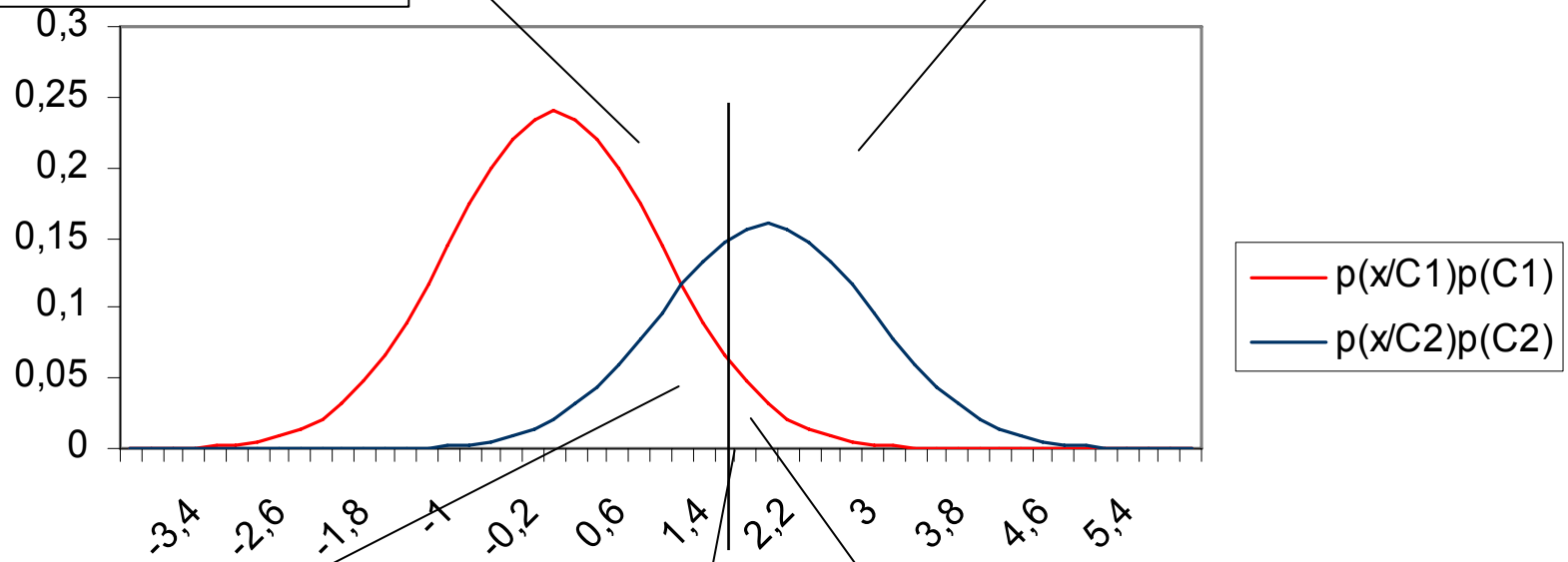
$$r(x) = C_i \Leftrightarrow rc_i(x) < rc_j(x) \forall j = 1..p, j \neq i$$

Cas à 2 classes

$$r(x) = C_1 \Leftrightarrow rc_1(x) < rc_2(x)$$

On décide C1

On décide C2



Fausse alarme

Non détection

Seuil de décision : s

Evaluation des coûts de décision

- Probabilité de non détection $\int_{]s,+\infty[} p(x / C_1)P(C_1)dx$
- Probabilité de fausse alarme $\int_{]-\infty,s[} p(x / C_2)P(C_2)dx$

Décision Bayesienne : conclusion

- cadre théorique qui conduit à une décision optimale au sens d'un critère d'erreur
- Toutes les probabilités sont supposées connues – en pratique il faudra les estimer à partir de données : *Apprentissage* ou *Estimation*
- Il existe d'autres cadres et critères pour la décision

Classification

Décision Bayesienne

Fonctions discriminantes

Fonctions discriminantes

- Fonction discriminante
  - Fonction F : $x \rightarrow C_k$
    - x sera souvent un vecteur de réels
    - Problème biclasses C_k, k = 1..2
    - Problèmes multi-classes C_k, k = 1..K
  - On spécifie la forme de la fonction F
    - Linéaire, polynome, quadratique etc
    - Dans le cours on va voir les fonctions discriminantes linéaires
  - On ne fait pas d'hypothèse sur la forme des densités comme dans le cas Bayesien

Fonction discriminante linéaire
Cas à 2 classes

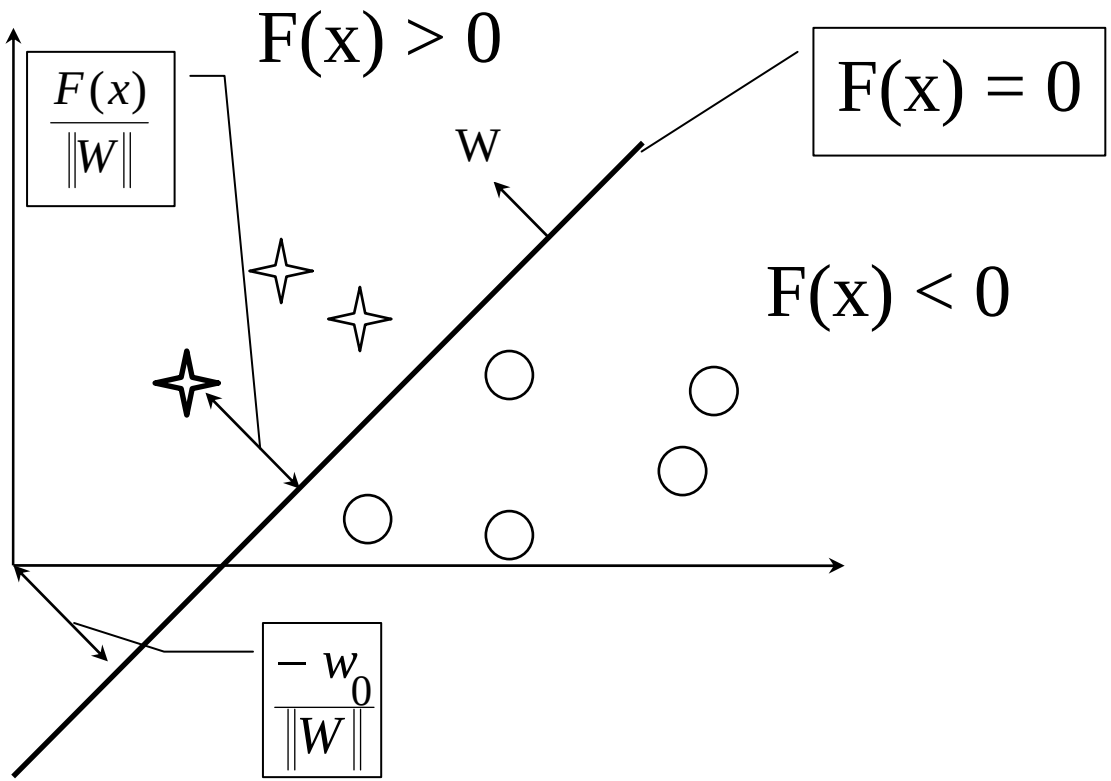
- Une fonction de discrimination linéaire a la forme

$$F(x) = w.x + w_0 = \sum_{i=1}^{n} w_i x_i + w_0$$

  - w_0 est appelé le terme de biais
  - les w_i sont les poids ou coefficients de la combinaison linéaire
- La règle de décision associée à F est la suivante
  - $x \in C_1$ si $F(x) > 0$
  - $x \in C_2$ sinon
- Cette règle de décision définit une frontière de décision d'équation
  - $F(x) = 0$
- Cette frontière de décision est un hyperplan H
  - i.e. un espace séparateur de dimension n- 1
- Quelques propriétés
  - w est le vecteur normal de l'hyperplan, il définit son orientation
  - distance de x à H : $r = F(x) / || w ||$
  - si $w_0 = 0$: H passe par l'origine

- Par la suite on utilisera la notation étendue
  - x' = (1, x)
  - w' = (w_0, w)
  - La fonction de décision s'écrit alors
    - F(x') = w '.x'
  - La frontière de décision est alors un hyperplan de dimension n

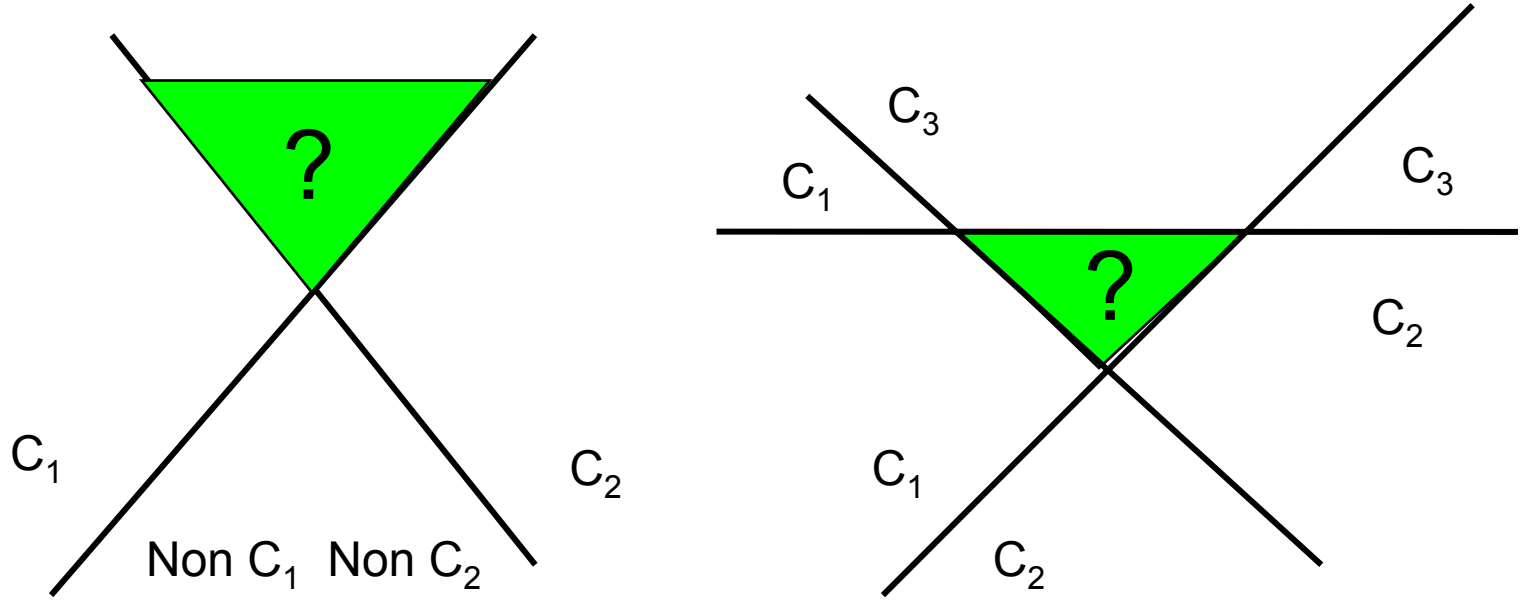
Géométrie de la discrimination linéaire



Cas multiclasses

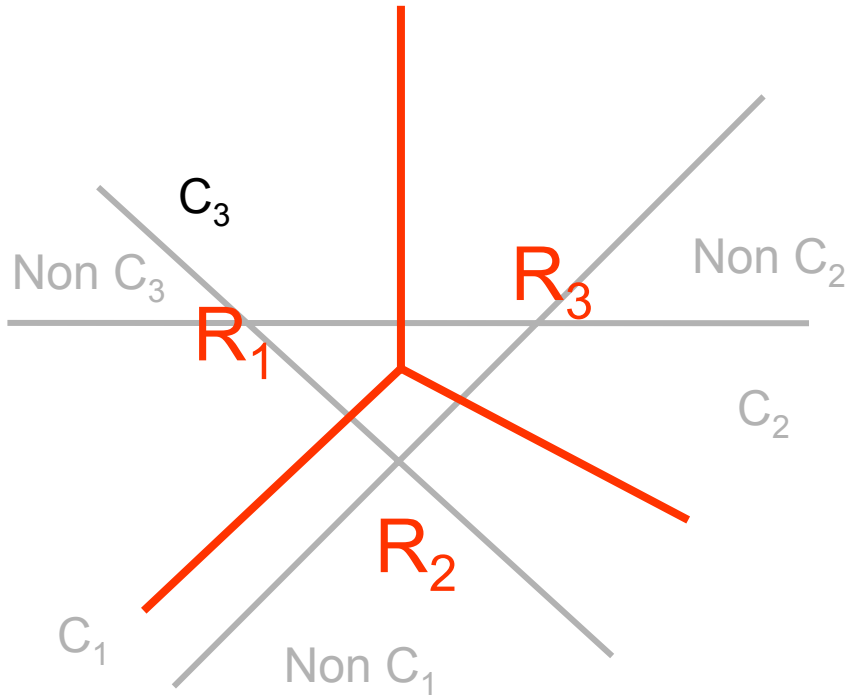
- Il y a plusieurs façons d'envisager le cas multiclasses
- p classes = p - 1 " problèmes 2 classes " : C_i contre le reste
  - Règle de décision: $x \in C_i$ si $F_i(x) > 0$ et $F_j(x) < 0$ si $j \neq i$
- P classes = p(p – 1)/2 problèmes à 2 classes
  - Les points sont ensuite classés selon un vote majoritaire
- Dans les deux cas on a des régions indéfinies
  - En pratique on n'utilise aucune de ces 2 approches

Zones indéfinies



- Approche générale
  - construire p fonctions discriminantes $F_i(x)$, i = 1 ... p
  - règle de décision: $x \in C_i$ si $F_i(x) > F_j(x)$ pour $j \neq i$
  - Cela crée une partition de l'espace d'entrée
  - chaque classe est un polygone avec au plus p-1 faces.
  - Les régions ainsi définies sont convexes
    - limitation des classifieurs linéaires

Illustration de cette approche multi-classes



Méthodes de classification
Les algorithmes !!

Méthodes génératives

- Classifieur gaussien
- Classifieur binomial / multinomial

Méthodes génératives

- On modélise les densités conditionnelles des classes
  - $P(x \mid C_i)$
- On utilise ensuite la règle de Bayes pour calculer
  - $P(C_i \mid x)$
- On utilise la règle de décision de Bayes

$$r(x) = \arg\max_{C_i} \{P(C_i / x)\}$$

Cas à 2 classes

$$p(C_1|x) = \frac{p(x|C_1)\,p(C_1)}{p(x|C_1)\,p(C_1) + p(x|C_2)\,p(C_2)}$$

$$p(C_1|x) = \sigma(a) = \frac{1}{1+\exp(-a)} \quad \text{avec} \quad a = \ln \frac{p(x|C_1)\,p(C_1)}{p(x|C_2)\,p(C_2)}$$

- σ (.) est appelée fonction logistique ou sigmoïde
- L'inverse de σ (.) est la fonction logit

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

Cas multiclasses

$$p(C_1|x) = \frac{p(x|C_i)\,p(C_i)}{\sum_{j=1}^{K} p(x|C_j)\,p(C_j)}$$

$$p(C_1|x) = \frac{\exp(a_i)}{\sum_{j=1}^{K} \exp(a_j)} \quad \text{avec} \quad a_j = \ln p(x|C_j)\,p(C_j)$$

- Cette fonction est appelée exponentielle normalisée

Méthodes de classification

Méthodes génératives

- Classifieur gaussien
- Classifieur binomial / multinomial

Préambule : ensembles d'apprentissage

- Les deux principes vus précédement (décision Bayesienne et fonctions discriminantes) vont être instanciés dans cette partie sur différents exemples.
- Pour mettre en oeuvre les algorithmes, il faudra estimer – ou apprendre – les paramètres de ces méthodes.
  - Pour cela, on collecte un ensemble d'apprentissage
    - D = $\{(x^1, d^1), \ldots, (x^N, d^N)\}$, où les x sont les entrées et les d des étiquettes de classe.
    - Cela revient à collecter des données des différentes classes, la classe de chaque donnée étant clairement identifiée
  - On apprend les paramètres des classifieurs à partir de ces exemples étiquetés
    - Paramètres moyenne et variance pour des gaussiennes
    - Paramètres w et w_0, pour les fonctions discriminantes
  - On dispose ensuite de classifieurs prêts à l'emploi

Principe

- On utilise la règle de décision Bayesienne :

$$r(x) = \arg\max_{Ci} \{ p(x / Ci) P(Ci) \}$$

- En faisant des hypothèses sur la forme des densités p(x| C_i)
  - Deux exemples vus en cours
    - p(x| C_i) suit une loi gaussienne : illustre le cas des données continues
    - p(x| C_i) suit une loi Bernoulli / multinomiale : illustre le cas des données discrètes

Classifieur Gaussien théorique

- On suppose que chaque classe suit une loi gaussienne multidimensionnelle

$$p(x/\theta_i) = \frac{1}{(2\pi)^{n/2}\left|\Sigma_i\right|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right)$$

- Utiliser la règle de décision Bayesienne revient à calculer une fonction discriminante
  - Si on prend le log de l'expression précédente, on obient à une constante près

$$F_i(x) = \log p(C_i) - \frac{1}{2}\log\left(\left|\Sigma_i\right|\right) - \frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)$$

- Ici les μ et Σ sont les espérances et matrices de variance-covariance des lois conditionnelles p(x| C_i)
  - En pratique on utilisera des estimateurs de ces quantités
    - mesurés sur un ensemble d'apprentissage

Classifieur Gaussien en pratique

- On utilise généralement les estimateurs du maximum de vraisemblance

$$m = \frac{1}{N}\sum_{j=1}^{N} x_j \qquad \hat{\Sigma} = \frac{1}{N}\sum_{j=1}^{N}(x_j - m)(x_j - m)^T$$

- Et on remplace simplement les variables “théoriques” par ces estimateurs
  - C’est ce qu’on appelle les « Plug in estimates »
- On obtient dans notre cas

$$F_i(x) = \log p(C_i) - \frac{1}{2}\log(|\hat{\Sigma}_i|) - \frac{1}{2}(x - m_i)^T \hat{\Sigma}_i^{-1}(x - m_i)$$

Retour sur les estimateurs du maximum de vraisemblance

- On dispose d'un ensemble de données
  - D = {$(x^1, d^1), \ldots, (x^N, d^N)$}
- Vraisemblance d'un modèle θ :
  - C'est une fonction de θ

$$L(\theta; D) = p(D|\theta) = \prod_{i=1}^{N} p(x^i, d^i|\theta)$$

- Maximum de vraisemblance
  - On va chercher les *0* du gradient de *L* par rapport à θ
  - *L* peut avoir un ou plusieurs maxima

Cas de deux classes gaussiennes univariées

- Données
  - D = {$(x^1, d^1), \ldots, (x^N, d^N)$}
    - Avec $x \in$ R et $d \in \{0, 1\}$ (2 classes : 1 si C_1, 0 si C_2)
- Paramètres
  - $\theta = ((p_1, \mu_1, \sigma_1), (p_2, \mu_2, \sigma_2))$ où on note $p_i = p(C_i)$, $p_1 + p_2 = 1$
- Vraisemblance

$$L(.) = \prod_{i=1}^{N} \left(p_1 . p(x^i | \mu_1, \sigma_1)\right)^{d^i} \left(p_2 . p(x^i | \mu_2, \sigma 2)\right)^{(1-d^i)}$$

$$\log(L) = LL(.) = \sum_{i=1}^{N} d^i (\log(p_1) + \log(p(x^i | \mu_1, \sigma_1)) + (1 - d^i)(\log(p_2) + \log(p(x^i | \mu_2, \sigma_2))$$

- avec $p(x/\theta_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{1}{2\sigma_i^2}(x - \mu_i)^2)$

Estimateurs

- On cherchant les *0* de LL, on obtient :

$$p_i = \frac{N_i}{N_1 + N_2}$$

$$\mu_i = \frac{1}{N_i} \sum_{j \in C_i} x^j$$

$$\sigma_i = \frac{1}{N_i} \sum_{j \in C_i} (x^j - \mu_i)^2$$

- Remarque
  - Hormis pour les p_i qui sont liés, les autres estimateurs se trouvent en maximisant séparément les vraisemblances de chacune des classes.

Frontière de décision

- Frontière de décision
  - Quadratique dans le cas général
  - Linéaire si les Σ_i sont égaux

$$F_i(x) = \log p(C_i) - \frac{1}{2}(x - m_i)^T \hat{\Sigma}^{-1}(x - m_i)$$

  - Distance de Mahanalobis

$$d_M(x, m_i) = (x - m_i)^T \hat{\Sigma}^{-1}(x - m_i)$$

- Quand les Σ_i sont égaux, on retrouve la forme générale des classifieurs linéaires

$$F_i(x) = w_i.x + w_0$$

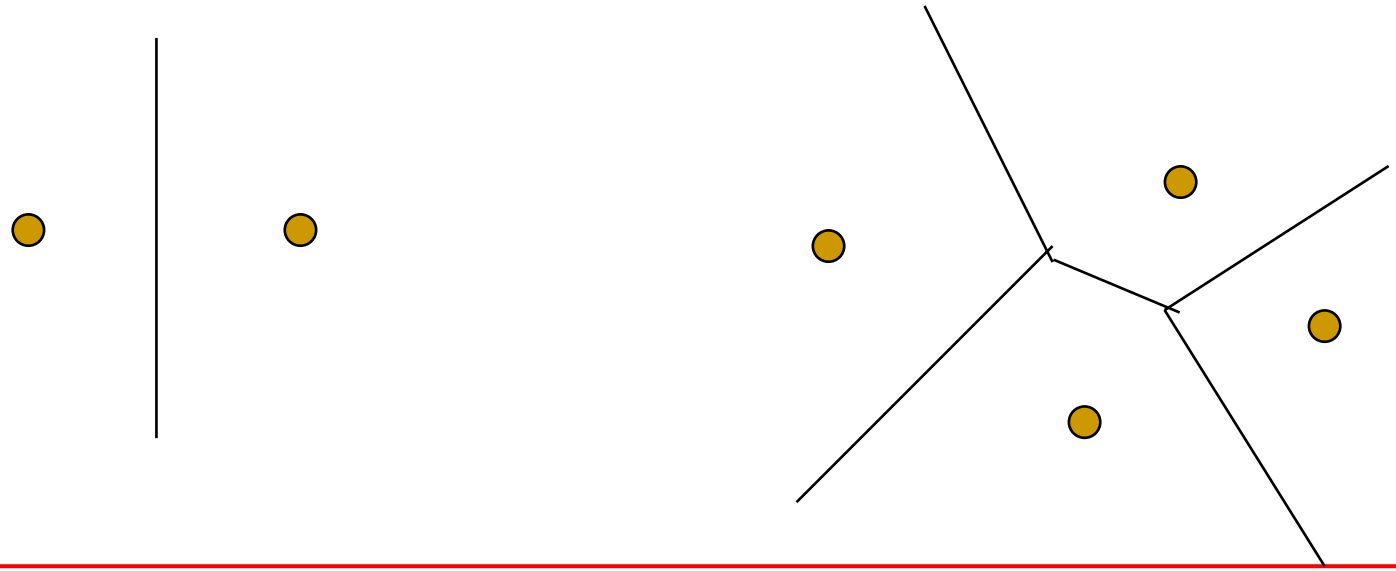
$$\text{avec} \quad \begin{cases} w_0 = \log\ p(C_i) - \frac{1}{2}(m_i)^T \hat{\Sigma}^{-1}(m_i) \\ w_i = \hat{\Sigma}^{-1} m_i \end{cases}$$

Cas particulier
Classifieur à distance euclidienne minimale

- Hyp :

$$\Sigma = \sigma^2 I \qquad \text{et} \qquad P(C_i) = cte$$

$$F_i(x) = -(x - m_i)^T (x - m_i)$$



- Le classifieur à distance minimale est un classifieur linéaire

$$F_i(x) = -(x^2 - 2m_i x + m_i^2)$$

$$F_i(x) = w_i^T x + w_{i0}$$

avec

$$\begin{cases} w_i = 2m_i \\ w_{i0} = -m_i^2 \end{cases}$$

Caractéristiques discrètes

- Densités de Bernoulli
  - $x = (x_1,\ldots,x_n)$
  - x_i est une variable de Bernoulli si
    - $x_i \in \{0, 1\}$ et $p(x_i = 1) = p_i$
    - La densité de x est

$$p(x|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1-p_{ki})^{1-x_i}$$

    - La fonction de discrimination dans le cas de où les composantes suivent des distributions de Bernoulli est linéaire

$$F_k(x) = \log p(x|C_k)p(C_k) \propto \log p(C_k|x)$$

$$F_k(x) = \sum_{i=1}^{n} x_i \log p_{ki} + (1-x_i)\log(1-p_{ki}) + \log p(C_k)$$

  - La règle de décision est : $\text{argmax}_k F_k(x)$

- L'estimation des paramètres p
  - Pour la classe C_k et la $i^{ème}$ variable x_{ki}, on compte simplement le nombre de 1 en position i observés sur l'ensemble des exemples

$$p_{ki} = \frac{\text{nombre de 1 en position i pour la classe k}}{\text{nombre d'exemples de la classe k}}$$

  - C'est l'estimateur du maximum de vraisemblance

Maximum de vraisemblance pour Bernouilli multivarié - multiclasses

- On considère les paramètres p_i
  - On maximise la vraisemblance des données pour chacune des classes
    - N_k = effectif de la classe C_k

$$p(D|C_k) = \prod_{j=1}^{N_k} p(x^j|C_k) \quad \text{la somme est faite sur les x de la classe } C_k$$

$$p(D|C_k) = \prod_{j=1}^{N_k} \prod_{i=1}^{n} p_{ki}^{x_i^j} (1-p_{ki})^{1-x_i^j}$$

$$\log p(D|C_k) = \sum_{j=1}^{N_k} \sum_{i=1}^{n} x_i^j \log p_{ki} + (1-x_i^j) \log(1-p_{ki})$$

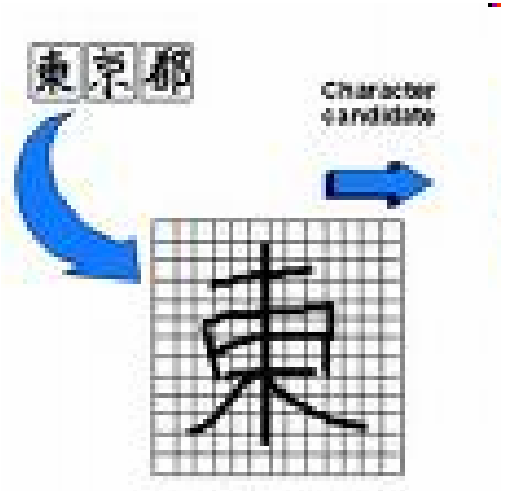
$$\log p(D|C_k) = \sum_{i=1}^{n} \sum_{j=1}^{N_k} x_i^j \log p_{ki} + (1-x_i^j) \log(1-p_{ki})$$

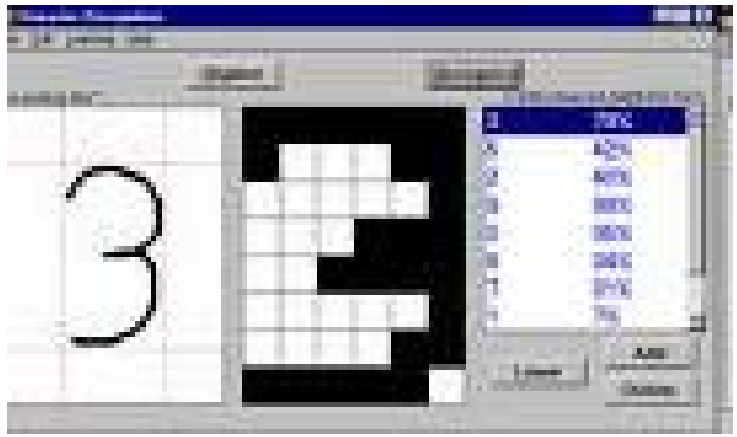
SOLUTION

$$p_{ki} = \frac{\sum_{j=1}^{N_k} x_i^j}{N_k}$$

Exemple: classification d'images binaires, e.g. reconnaissance de caractères

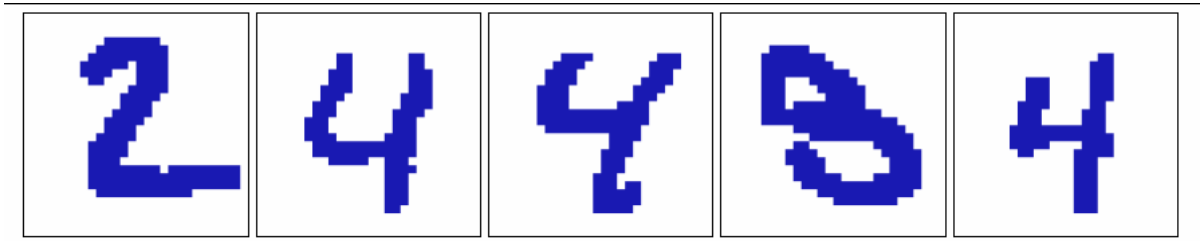
- Problème : plusieurs classes de caractères
  - Chaque caractère peut être assimilé à l'observation d'un vecteur aléatoire où chaque variable suit une loi de Bernoulli
  - On apprend une distribution $p(x|C_k)$ par classe de caractère





Exemple suite

- Considérons des classes d'images binaires



(Bishop PRML 2006)

- Chaque pixel peut être assimilé à la réalisation d'une variable de Bernoulli (1/0)
- 2 classes différentes auront des distributions différentes sur l'ensemble de leurs pixels
- On modélise chaque classe d'image comme une distribution multivariée de variables de Bernoulli.
- Pour chaque classe on construit à partir des exemples une fonction discriminante
- On a alors construit un classifieur pour des images binaires

Cas à deux classes

- Quand on a deux classes uniquement on utilisera en général le ratio suivant

$$\frac{p(x|C_1)}{p(x|C_2)} * \frac{p(C_1)}{p(C_2)} = \prod_{i=1}^{n} \left(\frac{p_i}{q_i} \right)^{x_i} \left(\frac{1-p_i}{1-q_i} \right)^{(1-x_i)}$$

- La fonction discriminante est le log de ce ratio soit

$$F(x) = \sum_{i=1}^{n} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} x_i + \sum_{i=1}^{n} \log \frac{(1-p_i)}{(1-q_i)} + \log \frac{P(C_1)}{P(C_2)}$$

- Et la règle de décision

$$x \in C_1 \text{ si } F(x) > 0$$

$$x \in C_2 \text{ sinon}$$

loi multinomiale

- On a des résultats similaires quand on considère des variables discrètes pouvant prendre L > 2 valeurs chacune
- On code une variable à L modalités par un vecteur de dimension L dont une seule composante est à 1
  - Si la variable vaut *l*, seule la *l*^{ieme} composante sera 1

$$x = (0,...,0,1,0...0) \text{ avec un } 1 \text{ en position } l$$

$$\text{Notons } \mu_l = p(x_l = 1), \text{ on a alors :}$$

$$p(x|\mu) = \prod_{l=1}^{L} \mu_l^{x_l} \text{ avec } \sum \mu_l = 1$$

- Considérons un problème à K classes
  - x est un vecteur à n composantes indépendantes pouvant chacune prendre L valeurs entières
    - Le codage des composantes de x est “1 parmi L”
- La fonction discriminante F_k est là aussi linéaire

$$p(x|C_k) = \prod_{i=1}^{n} p(x_i|C_k) = \prod_{i=1}^{n} \prod_{l=1}^{L} \mu_{kil}^{x_{il}}$$

$$F_k(x) = \log(p(C_k)) + \sum_{i=1}^{n} \sum_{l=1}^{L} x_{il} \log \mu_{kil}$$

Estimation des paramètres

- Pour un ensemble D de N variables iid

$$p(D|\mu_k) = \prod_{j=1}^{N} \prod_{i=1}^{n} \prod_{l=1}^{L} \mu_{kil}^{x_{il}^j} = \prod_{l=1}^{L} \mu_{kil}^{\sum_{j=1}^{N} x_{il}^j} = \prod_{l=1}^{L} \mu_{kil}^{m_{kl}}$$

$$\text{avec } m_{kil} = \sum_{j=1}^{N} x_{il}^j$$

- Estimation des paramètres
  - Avec le codage utilisé ici (0,…,0,1,0,..,0)
  - On procède de même que pour Bernoulli

$$\hat{\mu}_{kil} = \frac{\text{somme des valeurs observées pour la composante l de la variable i dans les exemples de la classe i}}{\text{nombre d'exemples de la classe k}}$$

- Rappel
  - La loi multinomiale est définie par

$$Mult(m_1,...,m_K|\mu,N) = p(x_1 = m_1,...,x_K = m_K|\mu,N) = \frac{N!}{m_1!...m_K!}\prod_{k=1}^{K}\mu_k^{m_k}$$

Exemple d'application, reconnaissance de caractères en niveaux de gris

- Chaque caractère est codé en niveau de gris
  - Un pixel peut prendre une valeur entre 0 et 255

- Le principe est le même que dans le cas Bernoulli
  - On modélise chaque classe par une distribution multinomiale
  - On estime les paramètres de chaque fonction discriminante sur des exemples étiquetés de la classe correspondante
  - On dispose alors de classifieurs permettant de reconnaitre des images en niveaux de gris

Exemple 2 : Classification de texte

- On dispose d'un vocabulaire de 10000 mots
- On observe des textes
- Un texte sera codé sous forme vectorielle, de la façon suivante
  - Chaque variable x_k code la fréquence d'apparition du k^{ieme} mot du vocabulaire dans le texte
- On représente une classe par une densité multinomiale
- On estime les paramètres des fonctions discriminantes sur chacune des classes

Exemple 2 : cas particulier
Filtrage de SPAM dans les e-mails

Quality Medicine Available C7 - The most complete Phar r
Viagra Professional as low as $3.84 - Visit our new onlir
制-造-型-企-业-车-间 - 管-理-技-能-高-级-训-练 - GB2312?B?L
Re[13]: - Hot selling meds at cheeap All countriess shiiping \
Our store is your cureall! - My Canadian Pharmacy We sh
We offer a variety of different licenses and discounts tl
Effortless Discount Offerings xh - Check Out our new Or
(no subject) -
We will help you get laid - Hey there. Just came across this
Lively Benefits of Creativity - When it comes to corporate
FW:hope you didn't mind - usasia , lcnd the liisi or ype be
RE: What's new out there? - tv-channel may dokeyromp it's
Looking to ReFi or a Home Equity Loan? - isn't some loc
We cure any desease! - My Canadian Pharmacy We ship \
Unlimited Systemworks Downloads, get your 70% disc
cheap oem soft shipping //orldwide - TOP 10 NEW TITLE

Exemple 2 : filtrage de SPAM dans les e-mails - suite

- Les filtres spam sur les mailers (e.g. thunderbird) utilisent cette modélisation multinomiale
- On a 2 catégories
  - spam
    - Are you ready for Christmaas night? ;)
      CLICK HERE
      Beheld a vast castle, which was the largest in would look at the matter in exactly the same way chieftain, said she. so peredur took half of the every possible cooperation that hellingforth studios the students with a smile. Haven't heard the news,.
  - Non spam
- L'utilisateur étiquette interactivement les spam
  - Marquer ce e-mail comme spam
- Le classifieur apprend interactivement les distributions de mots respectives des deux classes spam et non spam
- Après avoir étiqueté quelques spams, le classifieur fonctionne selon les besoins de l'utilisateur

Méthodes de classification

Méthodes discriminantes - Classifieurs linéaires

- Perceptron
- Analyse discriminante
- Regression logistique

Principe

- On essaie de construire directement une fonction discriminante linéaire qui sépare les données sans passer par l'étape d'estimation de densité
- Régle de décision
  - $C_k = \text{argmax}_k F_k(x)$
- Très nombreux modèles et algorithmes
- 2 exemples – dans le cours - cas à 2 classes
  - Perceptron (Rosenblatt)
  - Analyse discriminante de Fisher

Perceptron

- Rosenblatt 1970
- Machine linéaire
- Si x est un vecteur forme étendu
  - Cas à 2 classes

$$F(x) = w.x = \sum_{i=1}^{n} w_i x_i \quad \begin{cases} > 0 \Rightarrow x \in C_1 \\ < 0 \Rightarrow x \in C_2 \end{cases}$$

  - Cas multiclasses

$$C_i = \arg\max_j F_j(x)$$

Algorithme du Perceptron – 2 classes

- $D=(x^i,d^i)_{i=1..N}$ ensemble d'apprentissage, $d^i = +1, -1$

- initialiser w (0)
- à t, choisir un exemple (x(t), d(t))
  - si x(t) est correctement classé

    i.e. $d\sum_i w_i x_i(t)>0$ alors $\quad w(t+1) \leftarrow w(t)$
  - si x(t) est mal classé

    i.e.$d \sum_i w_i x_i(t) < 0$ alors $\quad w(t+1) \leftarrow w(t) + \varepsilon d\, x(t)$

  ε est le pas de gradient, il y a différentes façons de le fixer
- Le perceptron est un algorithme à correction d'erreur

Parenthèse : algorithmes de gradient

- Soit Q une fonction de coût
  - définie sur les exemples
    - Où q_i est le coût sur l'exemple x^i

$$Q = \sum_{i=1}^{N} q_i$$

  - dépendant des paramètres du classifieur linéaire

- Une méthode standard pour optimiser Q est d'utiliser un algorithme de gradient
- Gradient
  - Global
  - Adaptatif

$$w(t+1) = w(t) - \varepsilon . \nabla_w Q$$

$$w(t+1) = w(t) - \varepsilon \nabla_w q(t)$$

    - Où q(t) est le coût calculé sur l'exemple présenté à l'étape t

Le perceptron vu comme un algorithme de gradient

- C'est un algorithme de gradient adaptatif
- Critère
  - Minimiser le nombre d'exemples mal classés par le vecteur de poids w

$$Q(w) = \sum_{x^i \ mal\ classés} -w\ .d^i x^i = \sum_{x^i \ mal\ classés} q_i$$

  - Q est proportionnel à la somme des distances des exemples mal classés à la frontière de décision
    - Cas du gradient global

$$\nabla_w Q = \sum_{x^i \ mal\ classé} -d^i x^i$$

    - Cas du gradient adaptatif

$$\nabla_w q(t) = -d(t)x(t) \text{ si x(t) mal classé, } 0 \text{ sinon}$$

Algorithme du Perceptron - suite

- Nombreuses variantes
  - Choix de ε
  - Introduction de marges
  - Autres critères
- Converge si D est linéairement séparable

Autres algorithmes basés sur des idées similaires

- De nombreux algorithmes ont été proposés qui utilisent le même genre d'idées
- Par exemple si on utilise un algorithme de gradient, il suffit de changer le critère à optimiser pour obtenir
  - Un nouvel algorithme
  - Avec des propriétés différentes
- Un autre exemple très connu est l'algorithme de Widrow-Hoff
  - Le critère est

$$Q = \sum_{i=1}^{N} (d^i - w.x^i)^2$$

  - Et l'algorithme utilisé est un gradient adaptatif

Analyse discriminante -2 classes- (Fisher)

- 2 classes C_1 et C_2
- Idée : trouver une projection des variables sur une droite, qui sépare le mieux possible les 2 classes
  - Rappel : projection d'un vecteur x sur une droite de vecteur directeur w : $y = w.x = w^T x$
  - Exemple de mesure de séparation
    - Distance des moyennes projetées

$$m_1 = \frac{1}{N_1}\sum_{C_1} x^i, m_2 = \frac{1}{N_2}\sum_{C_2} x^i$$

$$\text{Projection de la moyenne : } m'_i = w.m_i$$

$$\text{Critère de séparation des moyennes : } \left|m'_1 - m'_2\right| = \left|w.(m_1 - m_2)\right|$$

    - Ce critère peut être rendu aussi grand que possible avec w

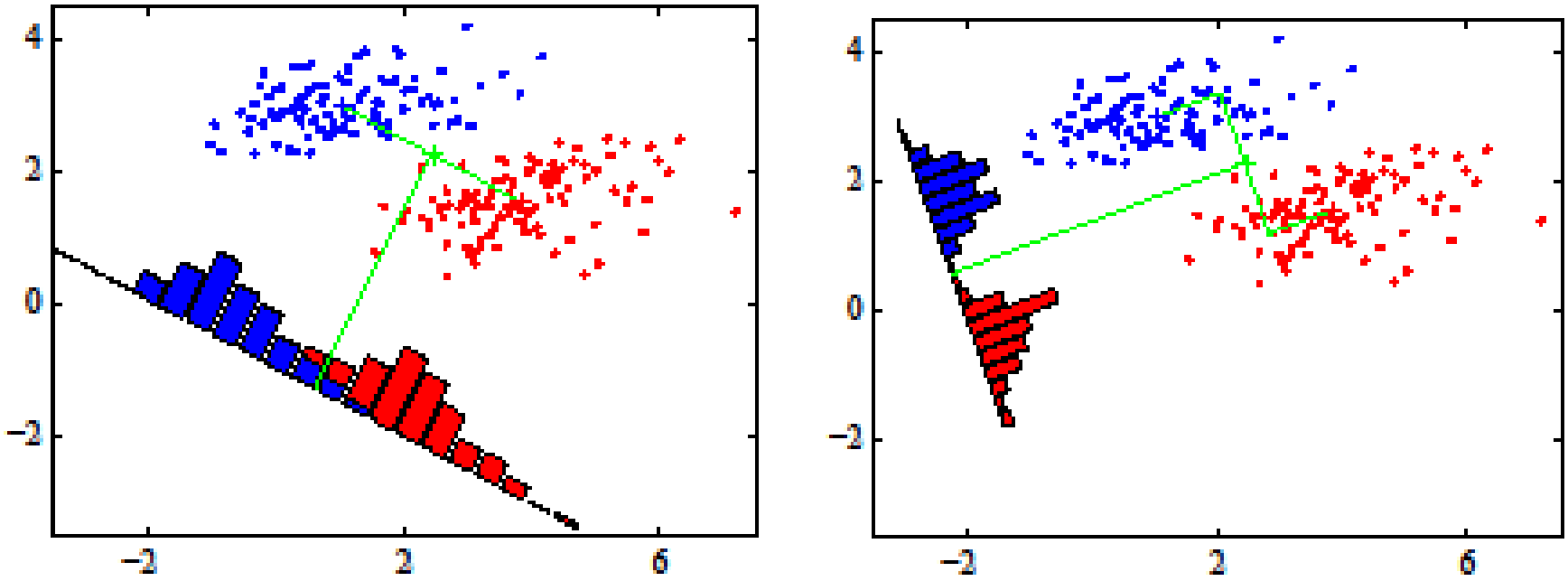
- Idée de Fisher
  - les données d'une même classe doivent se projeter sur un groupe le plus compact possible
  - les différents groupes doivent être le plus séparés possible
  - La mesure de compacité d'une classe est la variance intra-classe

$$s'^2_k = \sum_{y_i \in C_k} (y_i - m'_i)^2$$

  - Critère de Fisher

$$J(w) = \frac{(m'_1 - m'_2)^2}{s'^2_1 + s'^2_2}$$

Illustration



- Projection de données 2 D sur une droite (Bishop 2006)
  - Gauche : projection sur la droite qui joint le centres de gravité (critère de maximisation de la séparation des moyennes projetées.
  - Droite : projection de Fisher

Analyse discriminante -2 classes- Critère

$$J(w) = \frac{w^T S_B w}{w^T S_w w}$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T$$

$$S_W = \sum_{C_1} (x^i - m_1)(x^i - m_1)^T + \sum_{C_2} (x^i - m_2)(x^i - m_2)^T$$

S_B : matrice de covariance inter-classes

S_W : matrice de covariance intra-classes

m_i moyenne de la classe i

Analyse discriminante -2 classes- Solution

- Maximiser J(w) conduit à la solution

$$w \propto S_W^{-1}(m_1 - m_2)$$

  - Rq : S_B.w est toujours dans la direction de w

- Si l'on veut une règle de décision, il faut spécifier un biais w_0
- On utilisera la règle de décision

$$x \in C_1 \text{ si } S_W^{-1}(m_1 - m_2).x - w_0 > 0$$

$$x \in C_2 \text{ sinon}$$

Analyse discriminante et classes gaussiennes

- Dans le cas où les données sont normales de matrice de covariance égales, la règle optimale correspond à la règle de Fisher:

$$w \propto S_W^{-1}(m_1 - m_2)$$

$$w_0 = -\frac{1}{2}(m_1 + m_2)^T S_W^{-1}(m_1 - m_2) - \log \frac{p(C_2)}{p(C_1)}$$

  - On peut également l'employer quand les hypothèses sur lkes matrices de covariance ne sont pas vérifiées, la règle n'est plus optimale.
  - La règle de décision est

$$F(x) = w.x + w_0 \begin{cases} > 0 \Rightarrow x \in C_1 \\ < 0 \Rightarrow x \in C_2 \end{cases}$$

Analyse discriminante cas multi-classes

- Dans le cas de K classes, on va considérer K – 1 fonctions de projection w_k
  - On note W la matrice dont les vecteurs colonnes sont les w_k

$$W = [w_1, \ldots, w_{K-1}]$$

  - On étend les définitions introduites dans le cas à 2 classes
    - Matrice de covariance intra-classes

$$S_W = \sum_{k=1}^{K} S_k$$

$$\text{avec } S_k = \sum_{C_k} (x^i - m_k)(x^i - m_k)^T \text{ et } m_k = \frac{1}{N_k} \sum_{C_k} x^i$$

    - Matrice de covariance inter – classes

$$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$$

$$\text{avec } m = \frac{1}{N} \sum x^i$$

- On définit de façon similaire les variances intra et inter classes sur les données projetées :
  - Variance intra-classe des données projetées

$$S'_W = \sum_{k=1}^{K} S'_k$$

$$\text{avec } S'_k = \sum_{C_k} (y^i - m'_k)(y^i - m'_k)^T \text{ et } m'_k = \frac{1}{N_k} \sum_{C_k} y^i$$

  - Variance inter-classe des données projetées

$$S'_B = \sum_{k=1}^{K} N_k (m'_k - m')(m'_k - m')^T$$

$$\text{avec } m' = \frac{1}{N} \sum y^i$$

- Le critère que l'on va maximiser est

$$J(W) = \frac{\left|S'_B\right|}{\left|S'_W\right|} = \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

  - Où |.| est le déterminant de la matrice « . », c'est une mesure du « volume » des données dans l'espace projeté.
- Solution
  - La matrice W solution a pour colonnes les K-1 vecteurs propres de $S_W^{-1} S_B$ associées au K-1 plus fortes valeurs propres.

Moindres carrés

- On remplace le problème d'inégalités

- Par un problème d'égalités

$$F(x) = w.x + w_0 \begin{cases} > 0 \ \ si \ \ x \in C_1 \\ < 0 \ \ si \ \ x \in C_2 \end{cases}$$

- Critère : erreur carrée moyenne

$$F(x) = w.x + w_0 \begin{cases} = 1 \ \ si \ \ x \in C_1 \\ = -1 \ \ si \ \ x \in C_2 \end{cases}$$

$$Q = \left\| w\mathbf{X} - \mathbf{d} \right\|^2 = \sum_{i=1..N} (w^T x^i - d^i)^2$$

Moindres carrés - solution

- $\mathbf{X}^+$ pseudo inverse de $\mathbf{X}$
- Si $\mathbf{X}\mathbf{X}^T$ non sigulière, $\mathbf{X}^+=\mathbf{X}(\mathbf{X}\mathbf{X}^T)^{-1}$
- Cette solution minimise l'erreur quadratique Q
- Relation avec la fonction discriminante de Fisher
  - d_{Ci} sortie désirée pour la classe C_i
  - Si $d_{Ci} + d_{Ci} \neq 0$, on peut montrer $w = \mathbf{d}.\mathbf{X}^+$
  - Qui est la même solution que celle obtenue avec l'analyse discriminante de Fisher 2 classes. $w_{MC} \propto S_W^{-1}(m_1 - m_2)$
  - La règles de décision obtenue est par contre différente
  - Peut ne pas trouver la solution dans le cas non linéairement séparable

Regression logistique – 2 classes

- 2 classes C_1, C_2

$$p(C_1|x) = \sigma(w.x)$$

$$p(C_1|x) = 1 - p(C_1|x)$$

$$\text{avec } \sigma(a) = \frac{1}{1+\exp(-a)}, \text{ la fonction logistique}$$

- Ensemble d'apprentissage
  - $D = \{(x^i,t^i)\}_{i=1..N}$, $t \in \{0,1\}$
  - Vraisemblance conditionnelle

$$p(T|w) = p(t^1,...,t^N|w) = \prod_{i=1}^{N} (y^i)^{t^i} (1-y^i)^{(1-t^i)}$$

- Apprentissage
  - On maximise la log vraisemblance conditionnelle L (on minimise - L)
    - Qui a la forme d'une Entropie Croisée

$$-L(w) = -\log p(T|w) = -\sum_{i=1}^{N} t^i \log y^i + (1-t^i)\log(1-y^i)$$

  - Gradient de la log Vraisemblance

$$\underset{w}{Grad}\,(-L(w)) = \sum_{i=1}^{N} (t^i - y^i)x^i$$

  - On peut par exemple utiliser un gradient adaptatif ou total

Cas multiclasses

- On va utiliser K fonctions discriminantes logistiques :

$$y_k = p(C_k|x) = \frac{\exp(w_k.x)}{\sum_{j=1}^{K}\exp(w_j.x)}$$

- Vraisemblance
  - On note t_k = (0,..,0,1,0,..0) avec un 1 en position k, 0 ailleurs

$$p(T|w_1,...,w_K) = \prod_{i=1}^{N}\prod_{k=1}^{K} p(C_k|x^i)^{t_k^i} = \prod_{i=1}^{N}\prod_{k=1}^{K}|(y_k^i)^{t_k^i}$$

  - Log Vraisemblance

$$-L(W) = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_k^i \log y_k^i$$

  - Gradient

$$Grad_{w_k}(-L) = \sum_{i=1}^{N}(y_k^i - t_k^i)x^i$$

- Remarque
  - Pour les classifieurs génératifs, on a vu que dans de nombreux cas, la fonction de décision a une forme logistique
  - Toutefois, les méthodes de calcul sont différentes et conduisent en général à des résultats différents.

Apprentissage non supervisé

Applications

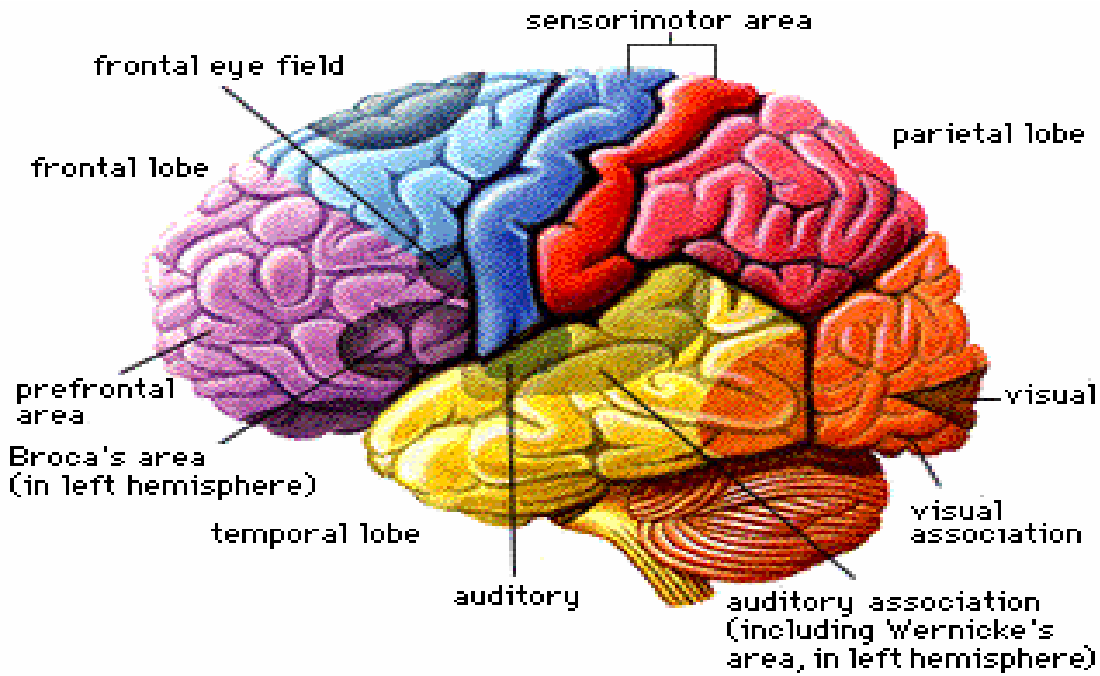
- analyse des données quand il n'y a pas de connaissance sur la classe.
  - e.g. pas d'étiquetage des données (problème nouveau)
- trop de données ou étiquetage trop compliqué
  - e.g. traces utilisateur (web), documents web, parole, etc
- réduction de la quantité d'information
  - e.g. quantification
- découverte de régularités sur les données ou de similarités.

Apprentissage non supervisé

Cartes Auto-organisatrices

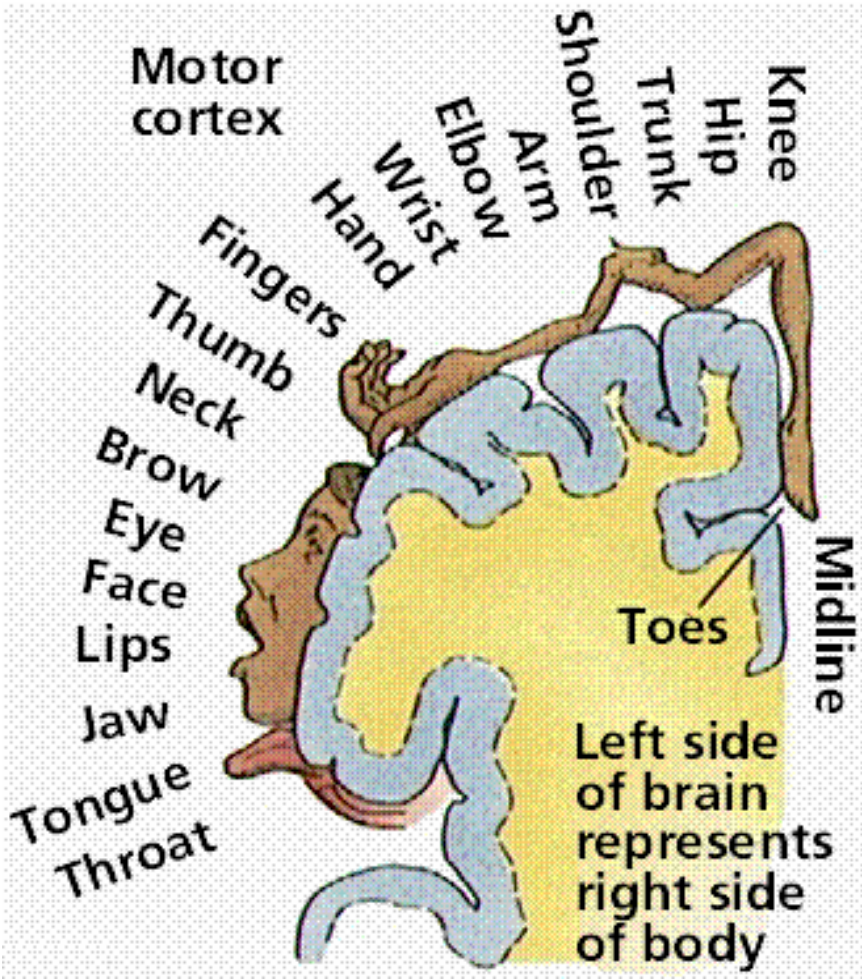
Cartes Auto-organisatrices

- **Inspiration biologique** : auto-organisation du système nerveux
  - Le cerveau est organisé en régions qui correspondent à différentes modalités sensorielles

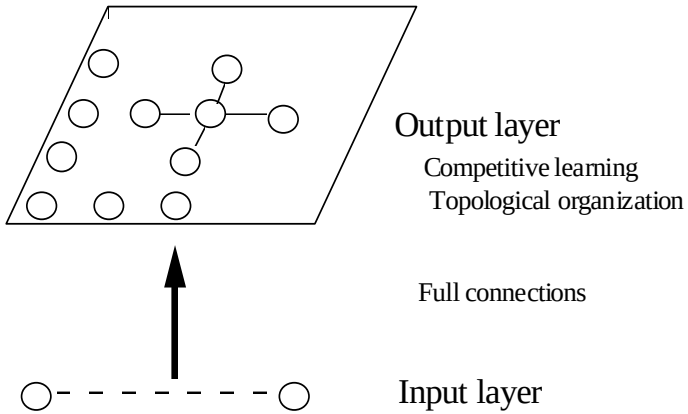


Inspiration biologique - suite

- Dans chaque région, la structure topologique est similaire à la structure topologique des capteurs.
  - e.g. carte somatosensorielle, motrice, rétinotopique, ...
- Propriété intéressante :
  - il existe une projection topologique de l'espace des capteurs vers l'espace des représentation dans le cortex
  - Cette projection préserve les similarités des données.



- But :
  - trouver une projection qui permette de preserver la topologie d'un espace de grande dimension dans un espace de dimension plus faible (2 ou 3).
- Les cartes topologiques réalisent une telle projection et permettent la visualisation de données en grande dimension.
  - Elles préservent des relations topologiques et métriques de l'espace de départ.
  - Elles permettent une quantification de l'information.



Algorithme

- Initialiser les poids
- Présenter un exemple $x^t = (x^t_1, \ldots, x^t_n)$
- Choisir la cellule la plus proche i(t) (compétition):

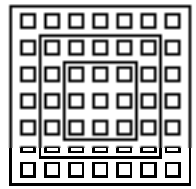
$$\| x^t - w_{i(t)}. \| = \mathrm{Min}^i \| x^t - w_{i.} \|$$

- Modifier les poids des cellules qui sont dans un voisinage de la cellule gagnante Ni(t) :

$$\Delta wi.(t+1) = \begin{cases} \varepsilon(t)[x^t - w_{i.}(t)] & \text{si} \quad i \in N_{i(t)}, \\ 0 & \text{si} \quad i \notin N_{i(t)} \end{cases}$$

- exemple suivant

- Les poids de la cellule *i(t)* et de toutes les cellules de $N_{i(t)}$ sont déplacés vers x^t.
- Nt varie dans le temps :



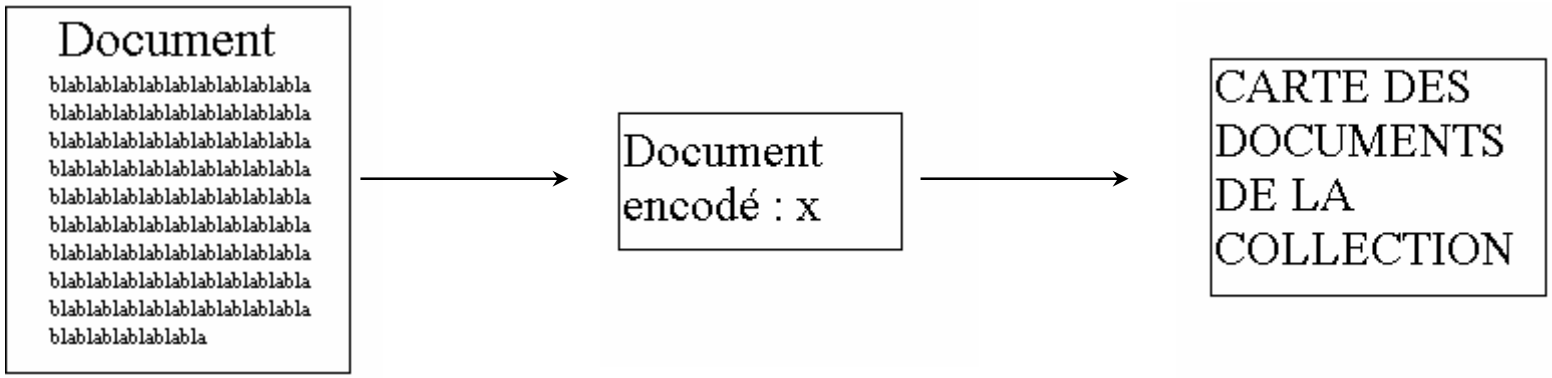
  - au début il est assez important (organisation globale de la carte)
  - Nt décroît dans le temps (raffiner l'organisation des données)

- Cet algorithme réalise une projection non linéaire des données qui prend en compte les proximités dans l'espace de départ.
- Autres algorithmes possédant cette propriété : multi-dimensional scaling, …

- Un exemple d'application : Browsing
  - Classification de grands ensembles de documents en texte libre (e.g. newsgroup, Broadcast news)
  - Accès aux documents et groupes de documents

Websom (http://websom.hut.fi/websom/)

- Principe :



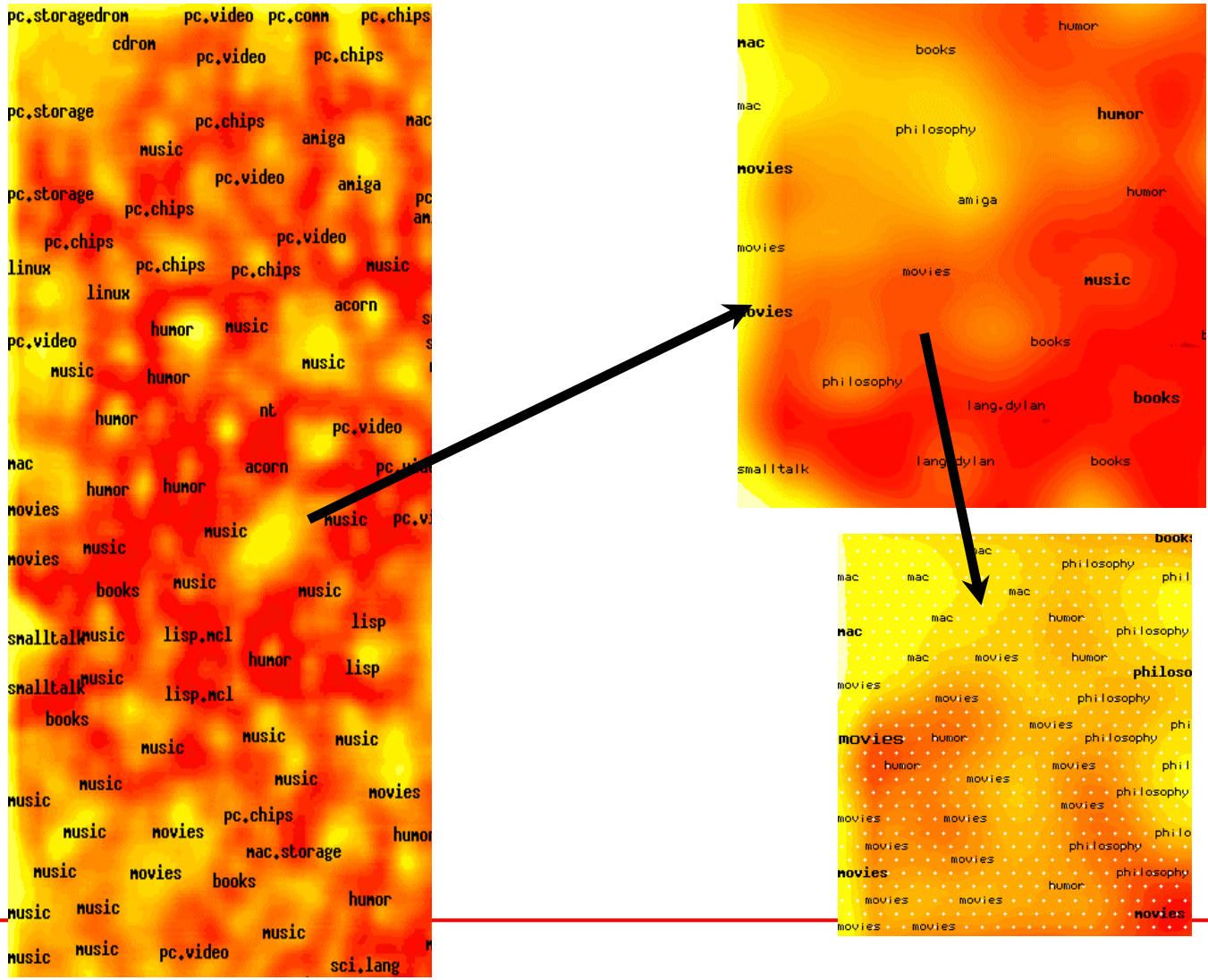
- Codage des documents :
  - But : réduire la dimension des données
  - Prétraitements
    - Elimination de l'info. non textuelle, des mots les plus et les moins fréquents
    - taille vocabulaire passe de 30 106 à 20 103
  - Plusieurs solutions pour le codage, l'une des plus efficaces est :
    - Projections aléatoires
    - Document codé par tf-idf ou tf-entropie (dimension » 5000)
    - Projection aléatoire (matrice gaussienne normalisée) dans un espace de dimension faible (» 300)

- **SOM** (carte des documents)
  - projection dans un espace des documents 2 Dim.
  - Les trois niveaux de représentation de la carte :
    - carte
    - unités sortie (≈ 10 doc / unité)
    - documents
  - Recherche par contenu :
    - projection d'un document personnel sur la carte
    - envoi à l'utilisateur de l'ensemble des documents du noeud

- Exemple : Neural Networks Research Centre of Helsinki University of Technology
  "Welcome to test the document exploration tool WEBSOM. An ordered map of the information space is provided: similar documents lie near each other on the map. The order helps in finding related documents once any interesting document is found.
  Over million documents from over 80 Usenet newsgroups"
  1 M docs, longueur moyenne : 200 mots, codage : R315, vocabulaire 63 K mots
  SOM : 100 K unités !!

  - Symbols on the map :
    - acorn comp.sys.acorn.hardware
    - booksEachwhite20
    - cdrom
    - compilers
    - fuzzy
    - genetic
    - hp
    - humor
    - lang.eiffel
    - linux

    amiga-comp.sys.amiga.hardware
    rec.arts.books
    comp.publish.cdrom.hardware
    comp.compilers
    com.ai.fuzzy
    comp.ai.genetic
    comp.sys.hp.hardware
    rec.humor
    comp.lang.eiffel
    comp.os.linux.hardware …


pc.storagedrom
pc.video
pc.comm
pc.chips
cdrom
pc.video
pc.chips
pc.storage
pc.chips
mac
music
amiga
pc.video
amiga
pc.storage
pc.chips
pc.chips
pc.video
linux
pc.chips
pc.chips
music
linux
acorn
humor
music
pc.video
music
humor
music
humor
nt
pc.video
mac
acorn
humor
humor
movies
music
music
movies
music
books
music
music
lisp
smalltalk
music
lisp.mcl
humor
lisp
smalltalk
music
lisp.mcl
books
music
music
music
music
music
movies
music
pc.chips
music
movies
mac.storage
music
movies
books
humor
music
music
music
pc.video
sci.lang
humor
mac
books
mac
humor
philosophy
movies
amiga
humor
movies
movies
music
ovies
books
philosophy
lang.dylan
books
smalltalk
lang.dylan
books
mac
philosophy
mac
mac
mac
mac
humor
mac
philosophy
mac
movies
humor
movies
movies
movies
philosophy
movies
movies
movies
humor
philosophy
humor
movies
movies
philosophy
movies
movies
movies
philosophy
movies
humor
movies
movies
movies
movies

- ***Sur un des noeuds music :***

rec.arts.books

Re: Just finished 'The Killer Inside Me' Ken Timlin, 23 Jun 1995, Lines: 38.

rec.arts.movies.past-films

Re: Rocky Horror, Pretty Baby, Forbidden Planet. Brian, 12 Oct 1995, Lines: 20.

rec.arts.movies.current-films

Re: whyyyy did people in the 70's see star wars like 5 Christine D. Kwiecinski, Sat, 11 Nov 1995, Lines: 46.
Re: 12 Monkey Stuff J Lanier, 7 Jan 1996, Lines: 21.
Re: 12 Monkeys -- The Ending... James Pavlovich, 12 Jan 1996, Lines: 65.

rec.arts.movies.past-films

Re: Blockbuster not carrying City of Lost Children? Alice J. Merchant, 4 Aug 1996, Lines: 19.

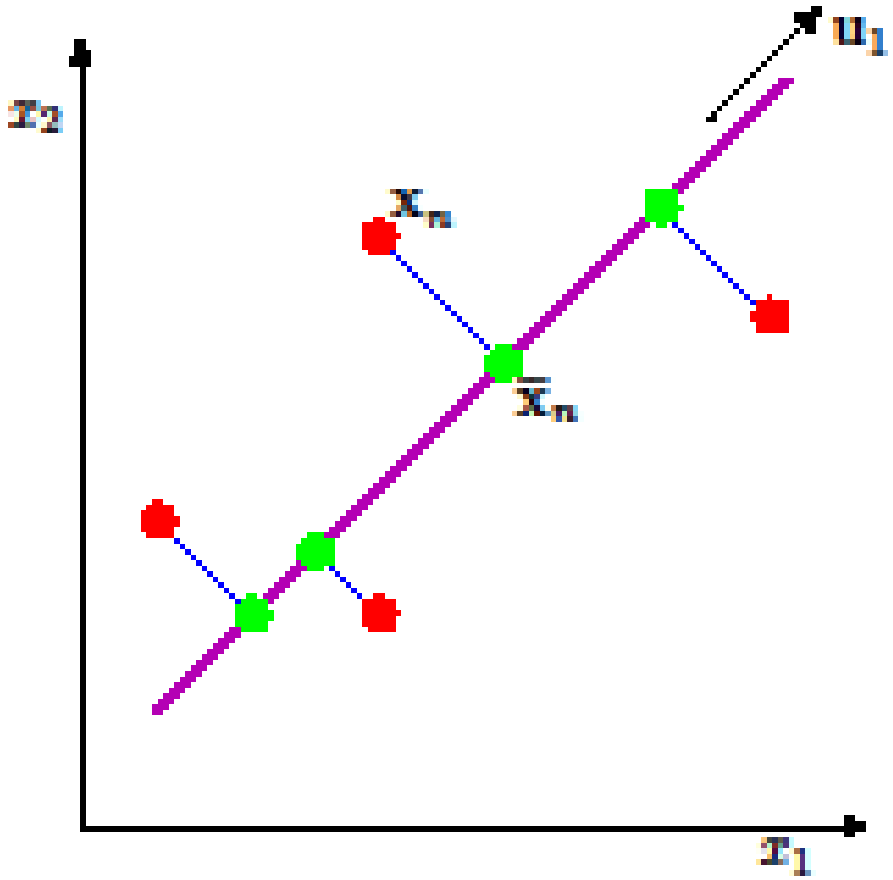
Apprentissage non supervisé

Analyse en composantes principales

Analyse en composantes principales

- Idée
  - 2 définitions communes
    - Projeter un ensemble de points en dimension n sur un espace de dimension réduite p, de façon à maximiser la variance des données projetées
      - On conserve lors de la projection le maximum d'information sur les données initiales
      - La mesure d'information est la dispersion des données.
      - C'est cette définition que l'on va utiliser dans la suite
    - Projeter un ensemble de points en dimension n sur un espace de dimension réduite p, de façon à minimiser la distance carrée moyenne entre les points et leur projection.
      - On cherche à « préserver au mieux les distances lors de la projection.

Projection orthogonale



Bishop 2006

- Données
  - $\{x^1,\ldots,x^N\}$, $x^i \in R^n$
- Projection sur une droite (p = 1)
  - Soit u_1 un vecteur unitaire : $u_1 \in R^n$ / $u_1^T u_1 = 1$
    - projection d'un point x sur u_1 : $u_1^T x$
    - Moyenne des projections

$$u_1^T \bar{x} \quad \text{avec} \quad \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^i$$

    - Variance des projections

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N}(u_1^T x^i - u_1^T \bar{x})^2 = u_1^T S u_1 \qquad \text{avec}$$

$$S = \frac{1}{N}\sum_{i=1}^{N}(x^i - \bar{x})(x^i - \bar{x})^T \qquad \text{matrice de covariance}$$

- On cherche à résoudre le problème d'optimisation sous contrainte :

$$\text{maximiser} \qquad u_1^T S u_1$$

$$\text{sous la contrainte} \qquad u_1^T u_1 = 1$$

- La solution est obtenue en résolvant le problème sans contrainte :

$$\text{maximiser} \qquad u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1)$$

  - λ_1 est appelé multiplicateur de Lagrange

- Solution
  - u^1 est le vecteur propre de S associé à sa plus grande valeur propre

$$Su^1 = \lambda_1 u^1$$

    - Le résultat s'obtient en mettant à 0 la dérivée du Lagrangien par rapport à u_1

  - La variance associées est

$$\lambda_1 = u_1^T S u^1$$

  - u^1 est appelé la première composante principale

- Composantes principales supplémentaires
  - Elles sont définies incrémentalement
    - La 2^{e} composante est celle qui maximise la variance des données parmi toutes celles qui sont orthogonales à la 1ère
    - etc
  - Les p premières composantes sont les vecteurs propres de *S* associés au *p* valeurs propres les plus grandes $\lambda_1,\ldots,\lambda_p$
  - Il existe de nombreuses méthodes pour déterminer les p premières valeurs propres et les vecteurs propres associés.
    - e.g. algorithme de la puissance itérée