INTRODUCTION À LA RECONNAISSANCE STATISTIQUE DES FORMES

Christophe AMBROISE

Mars 1997

CENTRE DE GÉOSTATISTIQUE
35, RUE SAINT-HONORÉ, 77305 FONTAINEBLEAU

ÉCOLE DES MINES
DE PARIS

Contents

Notations

Mathématique

$arg\max_t f(t)$	argument qui maximise la fonction f
$arg\min_t f(t)$	argument qui minimise la fonction f
C_N^x	coefficient binomial
$\mathbb{I}_A$	fonction indicatrice (vaut 1 si A est vrai, 0 sinon)
I	matrice identité
$f(t) \propto g(t)$	f et g sont proportionnelles

Statistique et reconnaissance des formes

$\mathbf{x}_i$	vecteur forme (typiquement $\mathbf{x}_i \in \mathbb{R}^p$)
$\mathbf{x}_{i,k}$	vecteur forme de la classe k
$\mathcal{C}_\infty, \cdots, \mathcal{C}_\mathcal{K}$	classes normales
$\mathcal{O}$	classe formée d'individus aberrants
$L(\theta, \hat{\theta})$	coût de choisir $\hat{\theta}$ à la place de $theta$ la valeur réelle
$R(k\|\mathbf{x})$	risque conditionnel : risque de décider de la classe k connaissant l'observation x
$R(\hat{c})$	risque total liée au classifieur $\hat{c}$, parfois appelé risque de Bayes
$f_k(\mathbf{x})$	densité de probabilité de la classe k
p_k	proportion de la classe k
$\boldsymbol{\mu}_k$	vecteur moyenne de la classe k
$\boldsymbol{\Sigma}_k$	matrice de variance covariance de la classe k
$\mathbf{m}_k$	centre de gravité de la classe k
$\mathcal{F}$	ensemble d'apprentissage
$D_\mathcal{F}(\hat{\theta})$	la déviance est définie comme deux fois la différence entre la log-vraisemblance du vrai modèle et celle du modèle de paramètre $\hat{\theta}$

Introduction

La reconnaissance des formes traite de la prise de décision automatique dans des problèmes de classement. De nombreuses méthodes de cette discipline trouvent leurs origines dans les statistiques. Ainsi des statisticiens tels que **?**) et **?**) se sont interessés à ce domaine, qui a réellement connu un essort important dans les années "soixante" poussé, par le développement de l'informatique. Analyse discriminante, discrimination, classement, apprentissage supervisé sont autant de termes différents qui désignent le problème de la reconnaissance des formes, que l'on pourrait définir de la façon suivante :

> À partir d'exemples de signaux complexes et de décision correctes concernant ces signaux, apprendre à prendre des décisions pour des signaux à venir.

C'est une activité que nous exercons tous les jours. Ainsi nous savons grâce à notre expérience passée :

- attribuer un nom à un visage connu,
- distinguer une voiture d'un camion,
- reconnaitre certaines espèces arbres,
- ...

D'un point de vue formel, ce type de problème peut s'exprimer comme suit :

- on considère des objets (individus), décrits par p caractéristiques qui définissent un vecteur forme $\mathbf{x}_i$ appartenant typiquement à un sous-ensemble de $\mathbb{R}^p$;
- ces objets appartiennent chacun à une classe parmi les classes $\mathcal{C}_\infty, \cdots, \mathcal{C}_\mathcal{K}$;
- le but est de classer un nouvel objet $\mathbf{x}$ dans l'une des K classes.

Souvent deux décisions supplémentaires sont introduites : le doute et le rejet d'individus aberrants ("outliers"). La décision de doute revient à se laisser la possibilité de ne pas

classer certains vecteurs forme. Décider de classer un individu comme n'appartenant à aucune classe est équivalent à considérer une classe $\mathcal{O}$, qui contient les individus aberrants (typiquement des erreurs de mesure).

Notons que l'école française d'analyse de données à la suite de Benzecri insiste sur la distinction entre les termes classement et classification. Un problème de classement ("classification" en anglais !) consiste à affecter des individus à des classes connues *a priori*. En classification ("clustering" dans le dialect anglo-saxon), on tente de découvrir une structure de classes qui soit "naturelle" aux données. Dans la littérature liée à la reconnaissance des formes, la distinction entre les deux approches est souvent désignée par les termes "apprentissage supervisé" et "non supervisé".

D'une manière générale, le modèle de base en reconnaissance des formes prend la forme suivante :

- extraction et sélection de caractéristiques,
- classement.

Les deux étapes sont complémentaires : si les caractéristiques extraites sont très discriminantes, la problème de classement devient évident. À l'inverse, si la méthode de classement est "infaillible", elle sera à même de classer sur la base de n'importe quel jeu de caractéristiques. En général le choix des caractéristiques extraites est plus lié au problème à résoudre que le choix de la méthode de classement.

Notons que les techniques d'extraction de caractéristiques sont parfois qualifiées de discrimination à but descriptif (**?**). En effet le but immédiat de l'extraction de caractéristiques n'est pas de classer de nouveaux individus mais de déterminer quelles variables ou combinaisons de variables caractérisent, décrivent, au mieux les différences entre les classes.

Exemple 0.1 (**?**) Hommes politiques et analyse linguistique :
Soit l'ensemble des députés élus en 1881. On désire discriminer deux groupes extrêmes :

- un groupe comprenant l'extrême gauche, les radicaux socialistes et les radicaux ;
- un autre groupe composé par les conservateurs, les conservateurs libéraux et les bonapartistes.

Il est possible de caractériser les hommes politiques par les fréquences de 53 mots (stéréotypes tels que "menace, famille, dieux, république...") figurant dans leurs discours. Dans ce cas, le but n'est pas de deviner la tendance d'un nouveau député mais plutôt de connaître quels sont les stéréotypes, les combinaisons de stéréotypes utilisés par les hommes politiques de chaque tendance.

△

Exemple 0.2 (**?**) On considère un ensemble de malades atteints du syndrome de Cushing (hypersécrétion de la glande adrenale). Trois causes possibles de la maladie sont connues. Pour chaque malade, on observe le taux de sécretion urinaire (mg/24 heures) de :

- tetrahydrocortisone,
- pregnanetriol.

Le problème consiste a évaluer la cause de la maladie de nouveaux patients dont on a mesuré les taux de sécrétion urinaire.

△

Il existe un grand nombre de méthodes qui visent à résoudre le problème de classement énoncé ci-dessus. Ces méthodes peuvent être partagées en deux approches :

- **approche statistique** ("sampling paradigm"), qui modélise dans un premier temps les densités $f_k(\mathbf{x})$, propres à chaque classe puis utilise les outils de la théorie de la décision statistique pour aboutir à une classification ;
- **approche discrimination** ("diagnostic paradigm"), qui vise à trouver directement les fonctions discriminantes, c'est à dire les fonctions qui permettent de décider quel individu appartient à quelle classe. C'est typiquement l'approche utilisée en discrimination linéaire ou bien par les techniques neuronales.

Les deux approches ne sont pas cloisonnées et sont équivalentes dans certaines circonstances. Dans tous les cas, le processus de classification vise à partitionner l'espace des caractéristiques en régions distinctes correspondant aux catégories.

Ces notes de cours visent à présenter un ensemble représentatif, mais non exhaustif, des techniques les plus usuelles en reconnaissance statistique des formes.

Les trois premiers chapitres présentent l'approche statistique. Le premier chapitre pose les fondements de la théorie de la décision utilisée en reconnaissance des formes et aborde les problèmes d'estimation des perfomances d'une procédure de classification. Le second chapitre est consacré aux méthodes paramétriques, c'est-à-dire aux méthodes issues de l'approche statistique qui font l'hypothèse que les densités $f_k(\mathbf{x})$ appartiennent à une famille donnée de distributions. Le troisième chapitre traite des méthodes non paramétriques qui ne posent pratiquement aucune hypothèse restrictive sur la forme des distributions.

Dans le chapitre quatre l'approche diagnostic est présentée. L'accent est mis en particulier sur la discrimination linéaire et sur les réseaux de neurones à couches.

Le cinquième chapitre introduit les méthodes non supervisées. Cette approche tient une place à part en reconnaissance des formes. Elle traite des problèmes où l'on recherche une structure de classe sans disposer d'exemples étiquettés. Nous concentrerons notre attention sur l'approche probabiliste en classification automatique.

Enfin le dernier chapitre adresse le problème particulier des données spatiales.

Chapter 1

Théorie bayésienne de la décision

L'estimation de quantités inconnues est un problème central des statistiques. Pour juger de la qualité d'une quantité estimée, la théorie de la décision statistique propose de considérer un critère. Habituellement, en théorie de la décision les critères utilisés sont basés sur des fonctions de coût :

$$L(\text{Valeur réelle}, \text{Décision})$$

qui mesurent le coût de prendre une certaine décision si la valeur réelle de la quantité à estimer est connue.

Exemple 1.1 Considérons le problème qui consiste à estimer le paramètre θ d'une densité de probabilité f, lorsque l'on dispose d'un ensemble $\mathbf{x} = (x_1, \cdots, \mathbf{x}_N)$ de réalisations indépendantes de cette loi. L'approche fréquentiste consiste alors à juger de la qualité de l'estimateur $\hat{\theta}$ de θ suivant le risque fréquentiste :

$$R(\theta, \hat{\theta}) = I\!E[L(\theta, \hat{\theta}(X))] = \int_{\mathcal{X}} L(\theta, \hat{\theta}(\mathbf{x})) f(\mathbf{x}|\theta) d\mathbf{x}, \tag{1.1}$$

où $L(\theta, \hat{\theta}(\mathbf{x}))$ est le coût de choisir $\hat{\theta}(\mathbf{x})$ alors que la valeur du paramètre est θ. Un risque classique est celui qui utilise un coût quadratique

$$L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2. \tag{1.2}$$

Dans ce cas, le risque peut être formulé comme la somme de la variance et du biais de l'estimateur :

$$R(\theta, \hat{\theta}) = I\!E[(\theta - \hat{\theta})^2] = Var[\hat{\theta}] + (I\!E_\theta[\hat{\theta}] - \theta)^2 \tag{1.3}$$

Ainsi le "meilleur" estimateur au sens de ce risque, parmi les estimateurs sans biais est celui qui possède la variance la plus petite.

△

Dans le cadre de la théorie bayésienne de la décision, la quantité à estimer est considérée comme une variable aléatoire et le critère de qualité pris en compte est le risque *a posteriori* (ou risque conditionnel), c'est-à-dire l'espérance du coût conditionnellement aux données observées :

$$R(\text{Décision}|\text{Données}) = I\!E[L(\text{Valeur réelle},\text{Décision})|\text{Données observées}]$$

Exemple 1.2 (suite de l'exemple 1.1) Ainsi, dans le contexte d'une approche statistique bayésienne, l'estimation d'un paramètre θ d'une distribution $f(\mathbf{x}|\theta)$, trois fonctions doivent être spécifiées :

1. la distribution a priori sur les paramètres, $\pi(\theta)$;
2. la loi sur les observations, $f(\mathbf{x}|\theta)$;
3. le coût associé à la décision $\hat{\theta}(\mathbf{x})$ pour le paramètre θ.

On appelle estimateur de Bayes associé à une loi a priori π et à un coût L, tout estimateur $\hat{\theta}^{\pi}(\mathbf{x})$ qui, étant donné un vecteur d'observation $\mathbf{x}$, minimise le coût *a posteriori*

$$I\!E[L(\theta, \hat{\theta}^{\pi}(\mathbf{x}))|\mathbf{x}] = \int_{\theta} L(\theta, \hat{\theta}^{\pi}(\mathbf{x}))\pi(\theta|\mathbf{x})d\theta.$$

△

Exemple 1.3 En géostatistique linéaire, le krigeage consiste à estimer en un point $\mathbf{x}$ la valeur prise par une fonction z dont on connaît les valeurs en N points $z(\mathbf{x}_1), \cdots, z(\mathbf{x}_N)$. En se plaçant dans le cadre de la théorie des variables régionalisées, on résout le problème en cherchant une estimation $z^*(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \cdot z(\mathbf{x}_i)$ qui minimise le critère :

$$I\!E[(Z^*(\mathbf{x}) - Z(\mathbf{x}))^2] \tag{1.4}$$

Dans le cas où l'on impose la contrainte de non biais, optimiser ce critère revient encore à chercher un estimateur de variance minimal.

△

Historiquement les travaux de Neyman et Pearson (1928) jetèrent les bases des tests d'hypothèses et donc de la théorie statistique de la décision. **?**) généralisa en introduisant les notions de risque et de coût. **?**) utilisa ce type d'approche décisionnelle dans le cadre de la reconnaissance des formes.

1.1 Discrimination et décision bayésienne

Dans le contexte de la discrimination, on dispose d'un ensemble d'observations $\{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}$ où $\mathbf{x}_i$ est un vecteur forme et $c(\mathbf{x}_i) \in \{1, \cdots, K\}$ est le label indiquant à quelle classe appartient le vecteur $\mathbf{x}_i$.

Le problème consiste à décider quelle classe attribuer à une nouvelle observation $\mathbf{x}$, c'est-à-dire à définir un estimateur $\hat{c}(\mathbf{x})$ de la classe $c(\mathbf{x})$.

Dans la suite de ce chapitre, nous noterons C l'indice de classe d'un vecteur forme aléatoire $\mathbf{X}$. Nous supposerons aussi connues les loi de probabilités relatives à toutes les variables aléatoires considérées. Cette hypothèse n'est évidemment pas réaliste mais permet de déterminer des stratégies de décision et de se faire une idée de la limite supérieure des performances de ces stratégies. Ainsi nous notons :

- les lois $P(\mathbf{X} = \mathbf{x}|C = k) = f_k(\mathbf{x})$ sur les vecteurs formes de chaque classe k,
- les probabilités *a priori* $P(C = k) = p_k$ de chaque classe,
- le coût $L(k, \hat{c}(\mathbf{x}))$ associé à la décision $\hat{c}(\mathbf{x})$ sachant que $C = k$.

Les probabilités *a posteriori* se déduisent des probabilités précédentes par la formule de Bayes :

$$\pi(k|\mathbf{x}) = \frac{p_k \cdot f_k(\mathbf{x})}{\sum_{\ell=1}^{K} p_\ell \cdot f_\ell(\mathbf{x})}$$

Dans ce cas le risque conditionnel associé à la décision $\hat{c}(\mathbf{x})$ est

$$R(\hat{c}(\mathbf{x})|\mathbf{x}) = I\!E[L(C, \hat{c}(\mathbf{x}))|\mathbf{x}] = \sum_{k=1}^{K} L(k, \hat{c}(\mathbf{x})) \cdot \pi(k|\mathbf{x}).$$

Au sens du risque conditionnel, la décision optimale $c^*(\mathbf{x})$ est celle qui vérifie :

$$c^*(\mathbf{x}) = arg \min_{\ell \in \{1,\cdots,K\}} \sum_{k=1}^{K} L(k, \ell) \cdot \pi(k|\mathbf{x}).$$

Remarquons que cette règle de décision minimise aussi le risque total :

$$R(\hat{c}) = I\!E\left[I\!E[L(C, \hat{c}(\mathbf{X}))|\mathbf{X}]\right] = \int_{\mathcal{X}} I\!E[L(C, \hat{c}(\mathbf{x}))|\mathbf{x}] \cdot f(\mathbf{x}) d\mathbf{x}.$$

En effet, comme $f(\mathbf{x})$ est toujours positive et que la règle de décision minimise le risque conditionnel pour chaque valeur de $\mathbf{x}$, elle minimise aussi le risque total. Cette règle est appelée la *règle de Bayes* et le risque total minimum $R(c^*)$ est appelé risque de Bayes.

1.1.1 Minimisation du taux d'erreur

Quelle fonction de coût L utiliser ? Une fonction particulièrement en faveur est la fonction de coût $\{0, 1\}$:

$$L(k, \ell) = \begin{cases} 0 \ si \ k = \ell, \\ 1 \ sinon. \end{cases}$$

Une bonne décision est gratuite (zéro) et une erreur coûte un. Toutes les erreurs sont pénalisées de la même manière et le risque conditionnel lié à ce coût s'exprime alors comme :

$$R(k|\mathbf{x}) = 1 - \pi(k|\mathbf{x})$$

Ce risque conditionnel peut s'interpréter comme la probabilité conditionnelle d'erreur de classement. La règle de Bayes consiste donc à choisir la classe la plus probable *a posteriori*,

$$c^*(\mathbf{x}) = arg \max_{\ell \in \{1, \cdots, K\}} \pi(\ell|\mathbf{x}).$$

ce qui revient, comme nous allons le montrer à minimiser la probabilité d'erreur de classement induite par le classifieur $\hat{c}$:

$$\begin{aligned} P(\text{Erreur}) &= \sum_{k=1}^{K} p_k P(\text{Erreur}|k), \\ &= \sum_{k=1}^{K} p_k P(\hat{c}(\mathbf{X})) \neq k|C = k), \end{aligned}$$

Considérons la décomposition du risque total sur les régions $\mathcal{R}_k$ où $\hat{c}(\mathbf{x}) = k$:

$$\begin{aligned} R(\hat{c}) &= \sum_{k=1}^{K} \int_{\mathcal{X}} P(\mathbf{X} = \mathbf{x}, C = k) \cdot L(k, \hat{c}(\mathbf{x})) d\mathbf{x}, \\ &= \sum_{k=1}^{K} \sum_{\ell=1}^{K} \int_{\mathcal{R}_\ell} p_k \cdot P(\mathbf{X} = \mathbf{x}|C = k) \cdot L(k, \hat{c}(\mathbf{x})) d\mathbf{x}, \\ &= \sum_{k=1}^{K} \sum_{\ell=1}^{K} p_k \cdot P(\hat{c}(\mathbf{x}) = \ell|C = k) \cdot L(k, \ell), \\ &= \sum_{k=1}^{K} p_k \cdot P(\hat{c}(\mathbf{x}) \neq \ell|C = k), \\ &= P(\text{Erreur}) \end{aligned}$$

Comme la règle de Bayes minimise le risque total, elle minimise aussi l'erreur de classement.

La partition de l'espace qui maximise la probabilité de prendre une décision correcte est bien celle obtenu par un classifieur de Bayes. Insistons sur le fait, que les remarques précédentes supposent bien évidemment que les distributions p_k et $f_k(\mathbf{x})$ sont toutes connues, alors qu'en pratique, l'on dispose seulement d'estimations.

Exemple 1.4 Illustrons de manière géométrique la minimisation de la probabilité d'erreur par la règle de Bayes correspondant au coût $\{0, 1\}$: considérons le cas limité à deux classes.

D'un point de vue géométrique, un classifieur partage l'espace en deux régions $\mathcal{R}_1$ et $\mathcal{R}_2$ et classe un vecteur forme dans la classe $\mathcal{C}_i$ si celui ci appartient à $\mathcal{R}_i$. Une erreur peut être commise de deux manières : ou bien une observation provenant réellement de la classe $\mathcal{C}_1$ tombe dans la région $\mathcal{R}_2$, ou bien une observation provenant réellement de la classe $\mathcal{C}_2$ tombe dans la région $\mathcal{R}_1$:

$$\begin{aligned} P(\text{Erreur}) &= P(\mathbf{X} \in \mathcal{R}_2, C = 1) + P(\mathbf{X} \in \mathcal{R}_1, C = 2), \\ &\quad P(\mathbf{X} \in \mathcal{R}_2 | C = 1) \cdot p_1 + P(\mathbf{X} \in \mathcal{R}_1 | C = 2) \cdot p_2, \\ &= \int_{\mathcal{R}_2} f_1(\mathbf{x}) \cdot p_1 d\mathbf{x} + \int_{\mathcal{R}_1} f_2(\mathbf{x}) \cdot p_2 d\mathbf{x}. \end{aligned}$$

Il est clair que le probabilité d'erreur sera minimale si lorsque $\mathbf{x} \in \mathcal{R}_2$, alors

$$\begin{aligned} f_1(\mathbf{x}) \cdot p_1 &< f_2(\mathbf{x}) \cdot p_2 \\ \pi(1|\mathbf{x}) &< \pi(2|\mathbf{x}) \end{aligned}$$

Cette règle de décision est exactement la règle de Bayes.

△

1.1.2 Introduction du doute

Introduire la notion de doute revient à considérer une décision supplémentaire :

$$\hat{c}(\mathbf{x}) = \mathcal{D}.$$

La stratégie classiquement retenue pour prendre en compte le doute utilise la fonction de coût suivante :

$$L(k, \ell) = \begin{cases} 0 \ si \ k = \ell, \ \text{(décision correcte)} \\ 1 \ si \ k \neq \ell \text{ and } \ell \in \{1, \cdots, K\}, \ \text{(erreur)} \\ d \ si \ \ell = \mathcal{D} \ \text{(doute)}. \end{cases} \tag{1.5}$$

La probabilité de doute liée à l'utilisation du classifieur $\hat{c}$ peut alors s'exprimer comme :

$$\begin{aligned} P(\text{Doute}) &= \sum_{k=1}^{K} p_k P(\text{Doute} | C = k), \\ &= \sum_{k=1}^{K} p_k P(\hat{c}(\mathbf{X}) = \mathcal{D} | C = k). \end{aligned}$$

Pour minimiser le risque total $R(\hat{c})$, il suffit de prendre la décision $c^*(\mathbf{x})$ qui minimise le risque conditionnel, qui sécrit

$$R(\hat{c}(x) = \ell|\mathbf{x}) = \sum_{k=1}^{K} L(k, \ell) \cdot \pi(k|\mathbf{x})$$

dans le cas général et vaut

$$\{\sum_{k=1}^{K} \pi(k|\mathbf{x})\} \cdot L(k, \ell) = d$$

si $\hat{c}(x) = \mathcal{D}$.

Suivant la décision considérée $\hat{c} = 1, \cdots, K, \mathcal{D}$, le risque conditionnel devient respectivement

$$\{1 - \pi(1|\mathbf{x}), \cdots, 1 - \pi(K|\mathbf{x}), d\}.$$

La décision prise, c'est-à-dire celle qui minimise le risque, est donc la suivante :

$$c^*(\mathbf{x}) = \begin{cases} k \ si \ \pi(k|\mathbf{x}) = \max_\ell \pi(\ell|\mathbf{x}) < (1 - d), \\ \mathcal{D} \ sinon. \end{cases}$$

Notons que pour qu'il existe une possibilité de doute il faut que :

$$0 \leq d \leq \frac{K-1}{K}.$$

En effet la somme des probabilités $\pi(k|\mathbf{x})$ valant 1, le maximum sur k de $\pi(k|\mathbf{x})$ est compris entre 1 et $\frac{1}{K}$ et l'on a :

$$0 \leq 1 - \max_k \pi(k|\mathbf{x}) \leq 1 - \frac{1}{K}.$$

Si l'on prend $d > \frac{K-1}{K}$, la règle de décision résultante correspond à la règle de Bayes pour le coût $\{0, 1\}$. Par contre, si d est choisi très petit, le classifieur doutera dans la plupart des cas. Ainsi, en pratique, le choix de la valeur de la constante d conditionne le comportement du classifieur.

Lorsque le doute est pris en compte par l'intermédiaire de la fonction de coût 1.5 alors le risque total peut s'exprimer comme une combinaison des probabilités d'erreur et de doute :

$$\begin{aligned} R(\hat{c}) &= I\!E[L(C, \hat{c}(\mathbf{X}))], \\ &= \sum_{k=1}^{K} \sum_{\ell=1}^{K} P(\hat{c}(\mathbf{X}) = \ell, C = k) \cdot L(k, \ell) + d \sum_{k=1}^{K} P(\hat{c}(\mathbf{X}) = \mathcal{D}, \mathcal{C} = \|), \\ &= \sum_{k=1}^{K} p_k \cdot P(\hat{c}(\mathbf{X}) \neq \ell | C = k) + d \cdot P(\text{Doute}), \\ &= P(\text{Erreur}) + d \cdot P(\text{Doute}) \end{aligned}$$

Dans ce cadre de décision statistique le problème central consiste donc à estimer les densités *a posteriori* $\pi(k|\mathbf{x})$ qui permettront de définir précisement le classifieur.

1.1.3 Traitement des données aberrantes

Les données aberrantes sont un problème délicat à traiter en discrimination. Une manière d'intégrer ce concept dans la théorie de la décision consiste à définir une classe spécifique $\mathcal{O}$ qui regroupe les données qui n'appartiennent à aucune autre classe. Un individu $\mathbf{x}$ sera alors classé donnée aberrante si :

$$p_{\mathcal{O}} f_{\mathcal{O}}(\mathbf{x}) \geq [d \cdot f(\mathbf{x}), \max_k p_k \cdot f_k(\mathbf{x})].$$

Cette règle revient à rejeter un vecteur forme $\mathbf{x}$ dans la classe $\mathcal{O}$ si $f(\mathbf{x})$ est trés petite. Notons malheureusement que cette démarche n'est envisageable que lorsqu'un nombre raisonnable d'individus aberrants (erreurs de mesure) est disponible. En effet, il faut que l'estimation de la distribution *a posteriori* relative à la classe des données aberrantes soit possible. C'est le cas dans certaines applications, comme la reconnaissance automatique des ZIP codes, qui donne de nombreux exemple d'erreurs de reconnaissance, mais dans la plupart des problèmes les données aberrantes sont rares et n'autorisent pas une estimation fiable des loi $p_{\mathcal{O}}$ et $f_{\mathcal{O}}(\mathbf{x})$.

Il est alors possible de raisonner directement sur la densité mélange et de considérer un seuil $s_{\mathcal{O}}$ tel que si :

$$f(\mathbf{x}) < s_{\mathcal{O}}$$

le vecteur $\mathbf{x}$ est considéré comme aberrant. Dans ce contexte, on pourra alors envisager de calculer ce seuil en fonction du pourcentage acceptable α d'individus aberrants :

$$P(f(\mathbf{X}) < s_{\mathcal{O}}) \leq \alpha.$$

1.2 Évaluation des performances

Comment évaluer les performances d'un classifieur de Bayes ? Si le coût $\{0, 1\}$ est utilisé, le taux d'erreurs de classement ($P(\text{Erreur})$) qui est le critère minimisé semble une mesure naturelle.

Si cette erreur de classement est estimée à l'aide de l'ensemble d'apprentissage, c'est-à-dire de l'ensembe $\{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}_N))\}$ qui a servi à définir le classifieur et dont on veut mesurer les performances, l'estimation risque de souffrir d'un biais optimiste quasi systématique. En effet, le classifieur a été défini pour minimiser le taux d'erreurs sur l'ensemble d'apprentissage et risque donc d'être plus performant sur cet ensemble que sur un autre ensemble de vecteurs forme. Cette estimation de l'erreur est couramment appelée erreur d'apprentissage.

1.2.1 Proportion de mal classés

La solution la plus simple pour estimer cette erreur de classement consiste à utiliser un ensemble de vecteurs forme dont on connaît la classe et qui n'a pas servi lors de l'apprentissage. Cet ensemble distinct est généralement appelé ensemble de test.

Si l'ensemble de test est constitué de M vecteurs forme et que le classifieur étudié commet R erreurs, on aura :

$$\hat{P}(\text{Erreur}) = \frac{R}{M}.$$

Cette approche possède deux inconvénients majeurs :

- il faut un ensemble de test "de grande taille" pour obtenir une estimation précise. Remarquons que le nombre d'erreurs R commises par le classifieur suit une loi binomiale $\mathcal{B}(M, P(\text{Erreur}))$. Un simple calcul d'intervalle de confiance montre que si pour avoir 95% de chances de connaître $P(\text{Erreur})$ à 1% près, il faut :

  $$M = P(\text{Erreur})(1 - P(\text{Erreur})) \cdot (1.96 \cdot 100)^2.$$

  Supposons que $P(\text{Erreur}) = 10\%$, il faudra alors un ensemble test de 3460 individus pour estimer à 1% de précision !

- il semble dommage d'utiliser un si grand nombre de vecteurs forme étiquettés dans le seul but de valider le classifieur alors que ceux-ci pourraient servir à améliorer la qualité de l'apprentissage.

1.2.2 Moyennage du risque

Pour améliorer la précision de l'estimation, il est possible de prendre en compte directement les probabilités *a posteriori*. Dans la section précédente nous avons considéré la variable aléatoire

$$R = \sum_{i=1}^{M} I\!I_{[C \neq arg\,\max_k P(k|\mathbf{X}_i)]}$$

comptant le nombre d'individus mal classés, ce qui revient à travailler directement sur la v.a.

$$Y = I\!I_{[C \neq arg\,\max_k P(k|\mathbf{X})]}.$$

Cet v.a. est un estimateur sans biais de la probabilité d'erreur. En effet, on a

$$I\!E[I\!I_{[C \neq argmax_k P(k|\mathbf{X})]}] = P(\text{Erreur}).$$

Sa variance s'exprime comme

$$\begin{aligned} var[Y] &= I\!E[Y^2] - I\!E[Y]^2 \\ &= P(\text{Erreur}) - P(\text{Erreur})^2 \end{aligned}$$

Considérons maintenant $Z = 1 - \max_k P(k|\mathbf{X})$ (notons que $1 - Z$ correspond au risque conditionnel minimisé par la règle de Bayes) comme estimateur de $P(\text{Erreur})$. Nous pouvons remarquer que c'est aussi un estimateur non biaisé :

$$I\!E[1 - \max_k P(k|\mathbf{X})] = R(\hat{c}) = P(\text{Erreur})$$

et que sa variance est plus petite que celle de l'estimateur précédent :

$$\begin{aligned} var[Z] &= I\!E[Z^2] - I\!E[Z]^2 \\ &\leq I\!E[Z \cdot (1 - \frac{1}{K})] - P(\text{Erreur})^2 \\ &\leq (1 - \frac{1}{K}) \cdot P(\text{Erreur}) - P(\text{Erreur})^2 \\ &\leq var[Y] - \frac{P(\text{Erreur})}{K} \end{aligned}$$

(Le passage à l'inégalité utilise le fait que $\max_k \pi(k|\mathbf{x}) \geq \frac{1}{K}$.) Un autre avantage de cet estimateur est qu'il ne repose pas sur la connaissance de la vraie classe des vecteurs formes de l'ensemble de test. Par contre il est évidemment très dépendant de la qualité de l'estimation des probabilités *a posteriori*.

1.2.3 Validation croisée

Dans le souci d'éviter d'utiliser des vecteurs forme spécialement pour l'estimation de la probabilité d'erreur, il est possible de recourir à la technique de validation croisée. Cette technique consiste à diviser l'ensemble des vecteurs formes étiquetés disponibles en V sous parties, et à définir le classifieur à l'aide de toutes les parties sauf une qui servira à l'estimation de la probabilité d'erreur. La procédure est répétée en utilisant successivement toutes les parties pour l'estimation, ce qui produit V différents taux d'erreur estimés. La version extrême de cette procédure, alors baptisée "leave one out", consiste à partager un ensemble de taille N en N sous parties.

Remarquons que l'estimation de cette probabilité peut se faire indifféremment suivant les deux techniques décrites précédement.

1.2.4 Méthode de réechantillonage : bootstrap

Une autre approche du problème de l'estimation de la probabilité d'erreur consiste à effectivement utiliser l'ensemble d'apprentissage qui produit un estimateur biaisé et à tenter d'estimer ce biais pour proposer un estimateur débiaisé.

Dans ce cadre, le but consiste donc à estimer le biais

$$B = I\!E[\hat{P}(\text{Erreur}) - P(\text{Erreur})].$$

Le bootstrap est une méthode adapté à l'estimation de ce biais, qui ne fait aucune hypothèse distributionnelle. Elle procède par réechantillonnage de l'ensemble d'apprentissage : de nouveaux ensembles d'apprentissage $\mathcal{F}_i^*$ de taille N sont réechantillonnés en utilisant un tirage avec remise parmi les N valeurs de l'echantillon initial $\mathcal{F}$. Pour chaque nouvel échantillon, le classifieur associé est calculé et deux taux d'erreur empiriques sont ainsi obtenus :

- $\hat{P}_i(\text{Erreur})$ sur $\mathcal{F}$ (l'ensemble initial),

- $\tilde{P}_i$(Erreur) sur $\mathcal{F}_i^*$ (l'ensemble généré par bootstrap),

Le biais est alors estimé par :

$$\hat{B} = I\!E_i[\hat{P}_i(\text{Erreur}) - \tilde{P}_i(\text{Erreur})]$$

où $I\!E_i$ est l'espérance sur tous les échantillons possibles. En pratique, $\bar{B}$ la moyenne empirique sur le plus grand nombre d'échantillons possible est utilisée pour estimer ce biais et l'estimation de l'erreur sera alors :

$$\hat{P}_{\text{bootstrap}}(\text{Erreur}) = \hat{P}(\text{Erreur}) + \bar{B}.$$

Cette procédure a le défaut d'être optimiste car les vecteurs forme de $\mathcal{F}$ utilisés pour le calcul du taux d'erreurs $\hat{P}_i$(Erreur) peuvent avoir été présents dans l'ensemble d'apprentissage $\mathcal{F}_i^*$. Efron a donc proposé de calculer les taux d'erreur $\hat{P}_i$(Erreur) en faisant intervenir les vecteurs forme de $\mathcal{F}$ qui ne sont pas dans $\mathcal{F}_i^*$. Ces taux d'erreurs sont ensuite moyennés pour obtenir ϵ_0 qui sert au calcul de l'estimateur du bootstrap dit ".632" :

$$\hat{P}_{\text{bootstrap632}}(\text{Erreur}) = 0.368 \cdot \hat{P}(\text{Erreur}) + 0.632 \cdot \epsilon_0.$$

"0.632"($\approx (1 - \frac{1}{e})$) n'est pas seulement un nombre magique mais aussi la probabilité lorsque N devient grand qu'un vecteur forme de $\mathcal{F}$ soit présent dans $\mathcal{F}_i^*$[1] !

1.2.5 Matrices de confusion

Pour caractériser les performances d'un classifieur, il semble aussi intéressant, en plus de l'estimation du taux d'erreur d'avoir une idée sur les classes qu'il confond. Les probabilités

$$e_{k\ell} = P(\hat{c}(X) = k | C = \ell)$$

permettent d'accéder à ce type d'information. La manière la plus simple d'estimer l'ensemble de ces probabilités qui forment ce que l'on appelle la matrice de confusion consiste à compter l'ensemble des décisions erronées pour chaque classe en utilisant un ensemble de test ou bien l'ensemble d'apprentissage.

[1] Notons K la v.a. comptant le nombre de fois où un vecteur de $\mathcal{F}$ apparait dans $\mathcal{F}_i^*$. On montre facilement que $P(K \geq 1) = 1 - P(K = 0) = (1 - \frac{1}{e})$, lorsque N tend vers l'infini.

Chapter 2

Approche statistique et modèles paramétriques

Dans le chapitre précédent nous avons vu que la règle de Bayes qui permet de définir un classifieur optimal au sens de la minimisation conjointe des probabilités d'erreur et de doute, nécessite la connaissance des loi *a posteriori* $\pi(\mathbf{x}|k)$. Malheureusement dans les problèmes réels de reconnaissance des formes, le modèle probabiliste que suivent les données est inconnu.

Dans le cadre de l'approche statistique ("sampling paradigm"), les densités considérées sont les densités Φconditionnelles aux classes $f_k(\mathbf{x}|\theta_k)$. En effet, la connaissance de ces densités conjointement aux proportions des classes p_k, suffit à appliquer la règle de Bayes car :

$$p_k \cdot f_k(\mathbf{x}|\theta_k) \propto \pi(k|\mathbf{x}).$$

Une procédure possible pour définir le classifieur consiste à considérer que les densités conditionnelles aux classes appartiennent à une famille de densités définies par peu de paramètres. Cette approche paramétrique comporte alors deux étapes :

1. le choix d'un modèle ;
2. l'estimation des paramètres de ce modèle.

Cette démarche revient à approximer les densités *a posteriori* par

$$\hat{\pi}(k|\mathbf{x}) = \frac{\hat{p}_k \cdot f_k(\mathbf{x}|\hat{\theta}_k)}{\sum_{\ell=1}^{K} \hat{p}_k \cdot f_\ell(\mathbf{x}|\hat{\theta}_\ell)}.$$

Ce chapitre présente dans une première partie quelques modèles couramment utilisés en reconnaissance statistique des formes. La seconde section expose l'estimation par maximum de vraisemblance des paramètres des densités de classe. L'estimation

par maximum de vraisemblance n'est naturellement pas la seule solution existante, qui permet de résoudre les problèmes d'estimation rencontrés en discrimination, mais elle a le mérite d'être relativement simple. Le lecteur intéressé pourra consulter l'excellent livre de B. Ripley(1996) pour une introduction à d'autres approches. Enfin, la dernière section envisage la délicate question du choix de modèle.

2.1 La loi normale multidimensionnelle

La loi normale repose sur les travaux de Jacques Bernouilli (1654-1705). Parfois attribué à Laplace et Gauss, elle tient son nom de Pearson (le père). F. Galton et K. Pearson utilisaient déjà la loi normale en dimension 2 à la fin du 19siècle et l'extension au cas général a été réalisé durant le premier quart du 20siècle.

Dans le cadre de la reconnaissance des formes, la loi normale est couramment utilisée comme densité de classe : on suppose ainsi que chaque $\mathbf{x}_i \in I\!R^d$ appartenant à la classe $\mathcal{C}_k$ suit une loi de densité

$$f_k(\mathbf{x}_i|\theta_k) = (2\pi)^{-\frac{d}{2}} \det|\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^t\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right) \tag{2.1}$$

avec $\boldsymbol{\mu}_k$ le vecteur moyenne, $\mathbf{\Sigma}_k$ la matrice de variance covariance, et d la dimension des vecteurs $\mathbf{x}_i$. De manière plus concise, on note

$$\mathbf{x}_i \sim \mathcal{N}_d(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$$

D'un point de vue géométrique (si l'on se place dans $I\!R^d$), faire l'hypothèse de normalité revient à supposer que tous les vecteurs forme d'une classe k donnée appartiennent à une hyperellipsoïde de centre $\boldsymbol{\mu}_k$ avec une certaine probabilité α. L'équation de l'hyperellipsoïde est donnée par :

$$r_\alpha^2 = (\mathbf{x}_i - \boldsymbol{\mu}_k)^t\mathbf{\Sigma}_k(\mathbf{x}_i - \boldsymbol{\mu}_k)$$

Si l'on décompose la matrice de covariance dans sa base de vecteurs propres :

$$\mathbf{\Sigma}_k = \lambda_k \cdot \boldsymbol{D}_k \cdot \boldsymbol{B}_k \cdot \boldsymbol{D}_k^t \tag{2.2}$$

il est possible d'interpréter géométriquement les quantités mises en évidence :

- $\lambda_k = \det|\mathbf{\Sigma}_k|^{\frac{1}{d}}$ est interprété comme le volume de la $k$classe. En effet plus λ_k est grand plus la classe occupera une place importante dans l'espace $I\!R^d$. Cette notion ne doit pas être confondue avec le nombre d'individus de la classe qui est relatif à la proportion p_k; ce n'est pas parce qu'une classe occupe un grand volume qu'elle contient forcément beaucoup d'individus.

- $\boldsymbol{B}_k$ (avec $det|B_k|$) est la matrice diagonale des valeurs propres. Elle caractérise la forme de la classe k. Plus une valeur propre est importante plus l'enveloppe de la classe est "allongée" dans la direction du vecteur propre correspondant.

- $\boldsymbol{D}_k$ est la matrice des vecteurs propres, qui sont les axes principaux de l'hyperellipsoïde. Cette matrice donne l'orientation de la classe k. C'est une matrice orthogonale de changement de base. Par rapport aux axes de référence, la base de vecteurs propres est obtenue par rotation.

Figure 2.1: Paramétrisation de la matrice de covariance dans le cas bidimensionnel

La matrice de covariance est symétrique définie positive. Dans $\mathbb{R}^d$, une matrice de covariance représente $d \cdot \frac{(d+1)}{2}$ paramètres à estimer pour une seule classe. Si l'on se place dans $\mathbb{R}^8$, cela fait 144 paramètres à estimer pour une seule matrice de covariance. Un moyen simple et classique de réduire ce nombre consiste à faire l'hypothèse (forte), que toutes les classes ont même matrice de covariance $\boldsymbol{\Sigma}$. Ainsi classiquement deux cas d'études sont distingués :

- hétéroscédasticité : matrice de covariance propre à chaque classe ($\boldsymbol{\Sigma}_k$).
- homoscédasticité : matrice de covariance commune à toutes les classes ($\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$).

2.1.1 Homoscédasticité

Lorsque les matrices de covariance sont identiques pour toutes les classes, cela revient à supposer que les vecteurs forme de chaque classe tombent dans des hyperellipsoïdes de même volume, même forme et même orientation. Dans ce cas la règle de Bayes choisit la classe $\hat{c}(\mathbf{x})$ telle que :

$$\begin{aligned}\hat{c}(\mathbf{x}) &= arg\max_k \pi(k|\mathbf{x}),\\ &= arg\max_k (p_k \cdot f_k(\mathbf{x})),\\ &= arg\min_k -2\cdot\log p_k + (\mathbf{x}-\boldsymbol{\mu}_k)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k).\end{aligned}$$

Remarquons que si les proportions p_k sont toutes égales, alors cette règle de décision revient à affecter un vecteur forme à la classe la plus proche au sens de la distance de **?**) :

$$\delta(\mathbf{x},\boldsymbol{\mu}_k) = [(\mathbf{x}-\boldsymbol{\mu}_k)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)]^{\frac{1}{2}}.$$

Si, de plus, la matrice de covariance est proportionnelle à la matrice identité,

$$\boldsymbol{\Sigma} = \sigma\cdot I,$$

alors la distance de Mahalanobis est équivalente à la distance euclidienne. Dans ce dernier cas les classes sont supposées avoir une forme sphérique et un volume σ. Lorsque les proportions sont différentes, le terme $-2\cdot\log p_k$ biaise la décision en faveur de la classe la plus probable *a priori*.

La règle de décision peut s'exprimer sous une forme plus simple lorsque le terme quadratique est développé, car $\mathbf{x}^t\boldsymbol{\Sigma}^{-1}\mathbf{x}$ est une expression indépendante de l'indice de classe :

$$\begin{aligned}\hat{c}(\mathbf{x}) &= arg\max_k (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k)^t\cdot\mathbf{x} + (-\frac{1}{2}{\boldsymbol{\mu}_k}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log p_k),\\ &= arg\max_k \mathbf{w}_k^t\cdot\mathbf{x} + w_{k0}.\end{aligned}$$

La fonction de décision est linéaire et on parle d'analyse discriminante linéaire, ce qui implique, que les frontières séparant deux régions voisines de décision, sont des hyperplans. Considérons $\mathcal{R}_k$ et $\mathcal{R}_\ell$ deux régions contigües : la frontière entre ces deux régions est décrite par l'équation :

$$(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k-\boldsymbol{\mu}_\ell))^t\cdot(\mathbf{x}-\mathbf{x}_0) = 0,$$

où

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_k+\boldsymbol{\mu}_\ell) - \log(\frac{p_k}{p_\ell})\frac{(\boldsymbol{\mu}_k-\boldsymbol{\mu}_\ell)}{(\boldsymbol{\mu}_k-\boldsymbol{\mu}_\ell)^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k-\boldsymbol{\mu}_\ell)}.$$

Ainsi la surface séparatrice est un hyperplan orthogonal à $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k-\boldsymbol{\mu}_\ell)$ et passant par le point $\mathbf{x}_0$.

Notons que dans le cas particulier où les proportions sont égales et la matrice de covariance proportionnelle à la matrice identité, alors l'hyperplan est orthogonal à l'axes reliant les vecteur moyennes $\boldsymbol{\mu}_k$ et $\boldsymbol{\mu}_\ell$ et le point $\mathbf{x}_0$ est exactement au milieu du segment défini par $\boldsymbol{\mu}_k$ et $\boldsymbol{\mu}_\ell$. Si les proportions sont différentes cela revient à translater l'hyperplan vers vecteur moyenne de la classe la moins probable.

Exemple 2.1 (**?**)(suite de l'exemple 1.4) Probabilité d'erreur dans le cas de deux classes gaussiennes sous hypothèse d'homoscédasticité : la règle de décision prend la forme suivante

$$\hat{c}(\mathbf{x}) = \begin{cases} 1 \ si \ A = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)) > \log \frac{p_1}{p_2}, \\ 2 \ sinon. \end{cases}$$

Si X appartient à la classe 1 alors on peut montrer que

$$A \sim \mathcal{N}(\frac{1}{2}\delta^2, \delta^2)$$

avec $\delta = [(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)]^{\frac{1}{2}}$. De même si X appartient à la seconde classe alors

$$A \sim \mathcal{N}(-\frac{1}{2}\delta^2, \delta^2)$$

Maintenant, la probabilité d'erreur peut s'écrire comme :

$$\begin{aligned} P(\text{Erreur}) &= P(\mathbf{X} \in \mathcal{R}_2 | C = 1) \cdot p_1 + P(\mathbf{X} \in \mathcal{R}_1 | C = 2) \cdot p_2, \\ &= p_1 \cdot P(A \leq \log \frac{p_1}{p_2} | C = 1) + p_2 \cdot P(A > \log \frac{p_1}{p_2} | C = 2), \\ &= p_1 \cdot \Phi(-\frac{1}{2}\delta + \frac{1}{\delta} \log \frac{p_1}{p_2}) + p_2 \cdot \Phi(-\frac{1}{2}\delta - \frac{1}{\delta} \log \frac{p_1}{p_2}) \end{aligned}$$

△

2.1.2 Hétéroscédasticité

Dans le cas général, chaque matrice de covariance est différente. La règle de décision prend une forme plus complexe que dans le cas précédent :

$$\begin{aligned} \hat{c}(\mathbf{x}) &= arg \max_k \pi(k|\mathbf{x}), \\ &= arg \min_k -2 \cdot \log p_k + \log(\det |\boldsymbol{\Sigma}_k|) + (\mathbf{x} - \boldsymbol{\mu}_k)^t \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k). \end{aligned}$$

Cette régle est quadratique et les surfaces de décisions générées sont des hyper-quadratiques : hypersphères, hyperellispoïde, hyperparaboloïde, hyperhyperboloïde. L'équation gérérale d'une surface séparatrice est alors de la forme :

$$\mathbf{x}^t W_k \mathbf{x} + \mathbf{w}_k^t \mathbf{x} + \mathbf{w}_{k0} = 0.$$

modèle	nombre de paramètres
$[\lambda DBD']$	$\alpha + \beta$
$[\lambda_k DBD']$	$\alpha + \beta + K - 1$
$[\lambda DB_k D']$	$\alpha + \beta + (K-1)(d-1)$
$[\lambda_k DB_k D']$	$\alpha + \beta + (K-1)d$
$[\lambda D_k BD'_k]$	$\alpha + K\beta - (K-1)d$
$[\lambda_k D_k BD'_k]$	$\alpha + K\beta - (K-1)(d-1)$
$[\lambda D_k B_k D'_k]$	$\alpha + K\beta - (K-1)$
$[\lambda_k D_k B_k D'_k]$	$\alpha + K\beta$
$[\lambda A]$	$\alpha + d$
$[\lambda_k A]$	$\alpha + d + K - 1$
$[\lambda A_k]$	$\alpha + Kd - K + 1$
$[\lambda_k A_k]$	$\alpha + Kd$
$[\lambda I]$	$\alpha + 1$
$[\lambda_k I]$	$\alpha + d$

Table 2.1: Nombre de paramètres à estimer pour chacun des quatorze modèles. Lorsque un paramètre différent est caclulé par classe, celui-ci est indicé. Si le paramètre est commun a toutes les classes, il ne porte pas d'indice. Nous avons $\alpha = Kd$ et $\beta = \frac{d(d+1)}{2}$

2.1.3 Modèle parcimonieux

Une solution intermédiaire pour réduire le nombre de paramètres à estimer consiste à considérer la paramétrisation de la matrice de covariance. Cette paramétrisation permet de mettre en évidence des paramètres qui possèdent une signification géométrique et l'on peut alors autoriser certains de ces paramètres a être commun à toutes les classes et estimer les autres par classe. Cette approche englobe bien évidemment les deux solutions présentées précédemment et permet de définir 8 modèles différents selon que l'on autorise la liberté par classe des paramètres de volume λ_k, de forme $\boldsymbol{B}_k$, ou bien d'orientation $\boldsymbol{D}_k$. Dans le contexte du classement, les modèles parcimonieux ont été exploités par Flury *et al.* (1994) et apppliqués entre autre par **?**) à la classification.

Pour diminuer encore le nombre de paramètres à estimer, il est possible d'envisager des hypothèses supplèmentaires sur les matrices de covariance. Deux situations semblent intéressantes :

- imposer le fait que les matrices de covariance sont diagonales. Dans ce cas,

  $$\boldsymbol{\Sigma}_k = \lambda_k \cdot_k$$

  avec $\det(_k) = 1$. Cette hypothèse fournit 4 modèles supplémentaires ;

- faire l'hypothèse que les matrices de covariances sont proportionnelles à la matrice identité :

  $$\boldsymbol{\Sigma}_k = \lambda_k \cdot I,$$

  ce qui ajoute encore deux modèles.

Finalement, nous avons quatorze modèles différent qui vont du plus simple forcant toutes les matrices de covariance à être proportionnelles à la matrice identité et à avoir même volume, au plus compliqué, qui nécessite le calcul d'une matrice de covariance différente par classe. Le tableau 2.1.3 donne le nombre de paramètre à estimer pour chacun des quatorzes modèles.

2.2 La loi de Student

?) remarque que la loi normale possède des "queues" très aplaties contrairement au distributions (empiriques) observées dans des problèmes réels. La loi de Student, qui possède des "queues" plus lourdes, peut donc être utilisée avantageusement.

La loi de Student multivariable peut être définie par la densité suivante lorsque ν le nombre de degré de liberté est supérieur à deux

$$f_k(\mathbf{x}|\theta_k) = \frac{\Gamma(\frac{1}{2}(\nu+d))}{(\nu\pi)^{\frac{d}{2}}\Gamma(\frac{1}{2}\nu)}|\boldsymbol{\Sigma}_k|^{-\frac{1}{2}}[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu}_k)^t\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)]^{-\frac{1}{2}(\nu+d)} \tag{2.3}$$

avec $\boldsymbol{\mu}_k$ le vecteur moyenne et $\boldsymbol{\Sigma}_k$ la matrice d'échelle. La matrice de covariance est $\frac{\nu\boldsymbol{\Sigma}}{\nu-2}$.

La règle de décision prend la forme

$$\hat{c}(\mathbf{x}) \quad = \quad arg\min_k \frac{\nu+d}{2}\log[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu}_k)^t\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)] + \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \log p_k$$

2.3 Estimation par maximum de vraisemblance

Entre autres contributions à la statistique, R. Fisher (1890–1962) a introduit le concept de vraisemblance en 1912 dans un article intitulé "On absolute criterion for fitting frequency curves" (**?**). Aujourd'hui (1997), les estimateurs du maximum de vraisemblance jouent un rôle central dans la théorie de l'estimation.

Définition 2.1 *Soit la réalisation d'un échantillon i.i.d.* $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ *d'une variable aléatoire de densité* f *dépendant d'un paramètre* θ. *On note* $f(\mathcal{X}|\theta) = \prod_{i=1}^N f(\mathbf{x}_i|\theta)$ *la densité de l'échantillon et* $\ell(\theta;\mathcal{X}) = f(\mathcal{X}|\theta)$ *la vraisemblance du paramètre* θ.

La notion de vraisemblance amène à réécrire la densité de l'échantillon en considérant le paramètre θ comme fonction d'un échantillon observé, $\mathcal{X}$. Cette définition du concept de vraisemblance conduit naturellement la définition d'estimateur du maximum de vraisemblance :

L'estimateur du maximum de vraisemblance $\hat{\theta}_{MV}$ de θ est tel que $\hat{\theta}_{MV} = arg\max_\theta \ell(\theta;\mathcal{X})$. Il est souvent plus avantageux de maximiser la log-vraisemblance $L(\theta;\mathcal{X}) = \log\ell(\theta;\mathcal{X})$ plutôt que la vraisemblance.

Dans le cas particulier où la log-vraisemblance est deux fois différentiable et le paramètre est un scalaire, $\hat{\theta}_{MV}$ est une solution du système :

$$\begin{cases} \frac{\partial L(\theta;\mathcal{X})}{\partial\theta} = 0 \\ \frac{\partial^2 L(\theta;\mathcal{X})}{\partial\theta^2} < 0 \end{cases}$$

Si le paramètre à estimer est vectoriel, l'estimateur du maximum de vraisemblance annule le gradient, et induit une matrice hessienne définie négative. Notons que ces conditions sont nécessaires mais pas suffisantes.

2.3.1 Discrimination et maximum de vraisemblance

Dans un problème de discrimination, l'on dispose d'un ensemble d'apprentissage

$$\mathcal{F} = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}_N))\}$$

Notons n_k le nombre de vecteurs forme de la classe k et $\mathbf{x}_{i,k}$ un vecteur forme de cette classe. La vraisemblance de l'ensemble des paramètres $(\theta, (p_1, \cdots p_K))$ du modèle

connaissant $\mathcal{F}$ s'écrit

$$\begin{aligned}\ell(\theta,(p_k);\mathcal{F})) &= \prod_{k=1}^{K}\prod_{i=1}^{n_k} P(\mathbf{X}_{ik}=\mathbf{x}_{i,k},C=k;\theta),\\ &= \prod_{k=1}^{K}\prod_{i=1}^{n_k} p_k\cdot f_k(\mathbf{x}_{i,k};\theta)\end{aligned}$$

Les estimateurs du maximum de vraisemblance de $(\theta,(p_1,\cdots p_{K-1}))$ maximisent

$$L(\theta,(p_k);\mathcal{F})) = \sum_{k=1}^{K}\sum_{i=1}^{n_k}\log f_k(\mathbf{x}_{i,k};\theta) + \sum_{k=1}^{K} n_k\log p_k. \tag{2.4}$$

Si nous cherchons dans un premier temps les estimateurs des proportions, en prenant en compte la contrainte $\sum_{k=1}^{K} p_k = 1$ grace à un multiplicateur de Lagrange, alors on trouve

$$\widehat{p}_k = \frac{n_k}{\sum_{\ell=1}^{K} n_\ell} = \frac{n_k}{N}$$

2.3.2 Les paramètres de la loi normale

Dans le cas où les densité de classe sont normales, les estimateurs du maximum de vraisemblance s'écrivent :

- pour les vecteurs moyennes

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n_k}\mathbf{x}_{i,k}}{n_k},$$

- et pour les matrices de covariance

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}(\mathbf{x}_{i,k}-\widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_{i,k}-\widehat{\boldsymbol{\mu}}_k)^t.$$

  Remarquons que cet estimateur est biaisé et qu'en pratique ce sont les estimateurs débiaisés qui sont couramment utilisés

$$\widehat{\boldsymbol{\Sigma}}_k^* = \frac{1}{n_k-1}\sum_{i=1}^{n_k}(\mathbf{x}_{i,k}-\widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_{i,k}-\widehat{\boldsymbol{\mu}}_k)^t$$

Si toutes les matrices de covariance des densité de classe sont supposées identiques (homoscédasticité) alors

$$\widehat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K}\frac{n_k}{N}\widehat{\boldsymbol{\Sigma}}_k$$

Ce dernier estimateur est aussi biaisé, car l'estimateur du vecteur moyenne est utilisé à la place du vecteur moyenne et l'on considère en général

$$\widehat{\mathbf{\Sigma}}^* = \sum_{k=1}^{K} \frac{n_k}{N-K} \widehat{\mathbf{\Sigma}}_k$$

qui est sans biais.

Lorsque la matrice de covariance est décomposée sur sa base de vecteurs propres (modèles parcimonieux), et que certains paramètres sont fixés le calcul des estimateurs du maximum de vraisemblance se complique légèrement et nécessite parfois l'usage d'algorithmes itératifs. Le lecteur intéressé consultera avantageusement Flury *et al.*(1994).

2.3.3 Les paramètres de la loi de Student

Dans le cas où les densités de classe sont de lois de Student multivariées, le calcul des estimateurs de maximum de vraisemblance devient itératif. Notons Q_{ik} le terme quadratique

$$Q_{ik} = (\mathbf{x}_{ik} - \boldsymbol{\mu}_k)^t \mathbf{\Sigma}_k^{-1} (\mathbf{x}_{ik} - \boldsymbol{\mu}_k),$$

et w_{ik} des coefficient de la forme

$$w_{ik} = \frac{1}{1 + Q_{ik}/\nu}.$$

En utilisant ces notations, les estimateurs de maximum de vraisemblance sont

- pour les vecteurs moyennes

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n_k} w_{ik} \mathbf{x}_{i,k}}{\sum_{i=1}^{n_k} w_{ik}},$$

- et pour les matrices $\mathbf{\Sigma}_k$ (qui ne sont pas les matrices de covariance, mais des matrices d'échelles).)

$$\widehat{\mathbf{\Sigma}}_k = \frac{1}{n_k} \frac{\nu + p}{\nu} \sum_{i=1}^{n_k} w_{ik} (\mathbf{x}_{i,k} - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_{i,k} - \widehat{\boldsymbol{\mu}}_k)^t.$$

Comme les coefficients w_{ik} sont fonctions des paramètres que l'on désire estimer, les estimateurs du maximum de vraisemblance ne peuvent être déterminés directement.

Remarquons que les estimateurs produits sont des versions pondérées des estimateurs trouvés dans le cas gaussien. Les pondérations donnant moins d'importance aux vecteurs forme isolés.

2.4 Choix de modèle

Lorsque un statisticien choisit un modèle, il sait pertinemment qu'il n'est qu'une approximation pratique de la réalité. Si la question n'est pas de trouver le modèle idéal que suivent les données, il semble par contre intéressant de déterminer quel est le meilleur modèle candidat parmi un certain nombre possible. Par exemple, on peut se poser la question de savoir quel est le modèle normal, parmi les quatorze modèles parcimonieux, qui rend le mieux compte de la densité d'une classe. On pourrait s'attendre à ce que le modèle le plus compliqué (celui qui possède le plus de paramètres) donne le meilleur résultat. Cette dernière observation se confirme lorsque l'ensemble d'apprentissage comporte de très nombreux exemples, mais ce n'est pas toujours le cas en pratique.

Mais comment évaluer les performances d'un modèle ? Dans le contexte de la discrimination, deux approches sont envisageable :

- une première solution consiste à envisager la question sous l'angle de la performance du classifieur produit par un modèle donné. Le modèle retenu sera celui qui engendre la plus petite probabilité d'erreur de classement. Cette perspective réduit alors le problème à la question traitée dans la dernière section du premier chapitre ;

- une seconde alternative consiste à considérer un critère d'ajustement du modèle à la réalité, classe par classe. Ainsi, le modèle le mieux ajusté devrait produire un classifieur le plus proche possible du classifieur bayésien idéal (celui qui classifie connaissant les densités réelles) et donc, indirectement minimiser l'erreur de classement. La fin de ce chapitre présente deux stratégies différentes permettant de choisir entre plusieurs modèles.

2.4.1 Pénalisation de la vraisemblance

Comment choisir entre plusieurs modèles de différentes complexités lorsque la méthode du maximum de vraisemblance est utilisé comme stratégie d'estimation ? En règle générale, sur un ensemble d'apprentissage donné, le modèle le plus complexe donne la plus grande vraisemblance. Une façon d'aborder le problème du choix consiste à tenter de répondre à la question suivante : comment se comporterait le modèle en moyenne si l'on jugeait ses performances avec d'autres donnéees ? L'idée sous-jacente est qu'un modèle très compliqué, ajusté en utilisant un petit ensemble d'apprentissage, risque de mal se comporter si l'on calcule ses performances sur un ensemble test, car le modèle sera "spécialisé" pour rendre compte du petit ensemble d'apprentissage.

La vraisemblance est une fonction d'un vecteur de paramètre θ, pour une certaine réalisation d'un échantillon. Si l'on écrit la log-vraisemblance en fonction d'un échantillon de taille N :

$$L_N(\theta) = \sum_{i=1}^{N} \log f(\mathbf{X}_i; \theta)$$

alors d'après la loi forte des grands nombres,

$$\lim_{N\to\infty} L_N(\theta)/N = \int f(\mathbf{x}) \log f(\mathbf{x};\theta) d\mathbf{x}.$$

Cette espérance possède souvent un maximum unique θ_0, qui peut être interprété comme la valeur de θ qui rend la densité $f(\mathbf{x};\theta)$ aussi proche que possible de la densité vraie $f(\mathbf{x})$ au sens de la distance de Kullback :

$$d(f, f_\theta) = \int f(\mathbf{x}) \log \frac{f(\mathbf{x})}{f(\mathbf{x};\theta)} d\mathbf{x}.$$

Sous certaines conditions, on montre que $\hat{\theta}$ l'estimateur du maximum de vraisemblance converge en loi vers θ_0.

Considérons maintenant l'espérance de l'écart au modèle, pour un seul vecteur forme :

$$D = 2I\!E[\log f(\mathbf{X}) - \log f(\mathbf{X};\hat{\theta})]$$

Cette valeur moyenne est intéressante car elle mesure l'ajustement moyen du modèle en prenant en compte d'autre vecteurs formes que ceux de $\mathcal{F}$. Le théorème suivant montre comment l'on peut estimer cette espérance et l'utiliser pour choisir le modèle le plus adapté.

Théorème 2.1 *(?)*

$$2I\!E[\log f(\mathbf{X}) - \log f(\mathbf{X};\theta_0)] = 2I\!E[\log f(\mathbf{X}) - \log f(\mathbf{X};\hat{\theta})] + \frac{1}{N} trace[KJ^{-1}] + O(1/\sqrt{N})$$

avec

- $$J = -I\!E[\frac{\partial^2 f(\mathbf{X}_i;\theta_0)}{\partial\theta\partial\theta^T}]$$
- $$K = var[\frac{\partial f(\mathbf{X}_i;\theta_0)}{\partial\theta}]$$

  la matrice d'information de Fisher.

Si l'on remplace espérance par moyenne empirique dans le théorème 2.1, alors on trouve

$$2\sum_{i=1}^{N} \log \frac{f(\mathbf{X}_i)}{f(\mathbf{X}_i;\theta_0)} \approx 2\sum_{i=1}^{N} \log \frac{f(\mathbf{X}_i)}{f(\mathbf{X}_i;\hat{\theta})} + trace[KJ^{-1}]. \tag{2.5}$$

Cette équation montre que la déviance de $\hat{\theta}$ calculée sur $\mathcal{F}$

$$D_{\mathcal{F}}(\hat{\theta}) = 2\sum_{i=1}^{N} \log \frac{f(\mathbf{X}_i)}{f(\mathbf{X}_i;\hat{\theta})},$$

gagne à être pénalisée par $trace[KJ^{-1}]$ pour approcher la déviance calculée sur un échantillon de taille infini et obtenir un critère de choix de modèle. Ainsi, l'équation 2.5 est à la base des critères de choix de modèle :

- NIC (Network Information Criterion) = $D_{\mathcal{F}}(\hat{\theta}) + 2d^*$,
  avec $d^* = trace[KJ^{-1}]$. On peut approximer les matrices K et J en remplacant espérance par moyenne sur $\mathcal{F}$ et θ_0 par $\hat{\theta}$.
- AIC (An Information Criterion)= $D_{\mathcal{F}}(\hat{\theta}) + 2d$;
  où d^* est remplacé par d le nombre de paramètre du modèle. Notons que cette approximation est justifiée lorsque la densité vraie $f()$ appartient à la famille f_θ, car dans ce cas $K = J$ et $d^* = trace[I]$.

Le modèle choisit par ce type d'approche est bien évidemment celui qui donne la plus petite valeur du critère choisit. En terme de vraisemblance, cela revient à choisir le modèle qui maximise la vraisemblance pénalisée par un terme négatif, lié au nombre de paramètres du modèle.

2.4.2 Sélection par validation croisée

Il est aussi possible de recourir à une procédure de validation croisée pour choisir le meilleur modèle au vue d'un certain critère. Si le critère considéré est la déviance alors cela revient à utiliser le critère NIC.

Supposons que l'on dispose d'un échantillon $\mathcal{F} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$ et notons $\mathcal{F}_{\setminus i}$, l'échantillon $\mathcal{F}$ sans $\mathbf{x}_i$. La procédure de validation croisée consiste à estimer $\hat{\theta}_i$ par maximum de vraisemblance en utilisant $\mathcal{F}_{\setminus}$ et à calculer le terme

$$D_i(\hat{\theta}_i) = 2(\log f(\mathbf{x}_i) - \log f(\mathbf{x}_i; \hat{\theta}_i))$$

La validation croisée de l'écart au modèle s'exprime alors comme la somme de tous les D_i obtenus. Si l'on considère le developpement de taylor de cette somme à l'ordre 1 en $\hat{\theta}$, on trouve :

$$\begin{aligned}
\sum_{i=1}^{N} D_i(\hat{\theta}_i) &= \sum_{i=1}^{N} D_i(\hat{\theta}) + \sum_{i=1}^{N} (\hat{\theta}_i - \hat{\theta})^t D_i'(\tilde{\theta}_i) \\
&= D_{\mathcal{F}}(\hat{\theta}) + \sum_{i=1}^{N} (\hat{\theta}_i - \hat{\theta})^t D_i'(\tilde{\theta}_i)
\end{aligned}$$

avec $\tilde{\theta}_i$ une combinaison convexe de $\hat{\theta}_i$ et $\hat{\theta}$. Considérons ensuite le développement de taylor du vecteur $D_i'(\hat{\theta}_i)$ (qui est le transposé du gradient de $D_i(\hat{\theta}_i)$ par rapport à $\hat{\theta}_i$) :

$$\begin{aligned}
D_i'(\hat{\theta}_i) &= D_i'(\hat{\theta}) + (\hat{\theta}_i - \hat{\theta})^t D_i''(\overline{\theta}_i) \\
D_i'(\hat{\theta}_i) &= (\hat{\theta}_i - \hat{\theta})^t D_i''(\overline{\theta}_i)
\end{aligned}$$

avec $\overline{\theta}_i$ une combinaison convexe de $\hat{\theta}_i$ et $\hat{\theta}$. Si tous les estimateurs converge vers θ_0, alors

$$\begin{aligned}
\sum_{i=1}^{N} D_i(\hat{\theta}_i) &\approx D_{\mathcal{F}}(\hat{\theta}) + \sum_{i=1}^{N} D_i'(\theta_0)^t D_i''(\theta_0)^{-1} D_i'(\theta_0) \\
&\approx D_{\mathcal{F}}(\hat{\theta}) + trace[KJ^{-1}].
\end{aligned}$$

Ce qui montre que cette approche est asymptotiquement équivalente au choix de modèle sur la base du critère NIC.

Chapter 3

Approche statistique et méthodes non paramétriques

Les méthodes non paramétriques d'estimation de densité, ne font pratiquement aucune hypothèse sur la forme de la densité. Utilisées dans le contexte de la discrimination, elles peuvent servir à obtenir des estimations de densités de classe. Si les p_k sont estimés sur la base des proportions observées dans l'ensemble d'apprentissage, les loi *a posteriori* sont alors approch'ees par :

$$\hat{\pi}(k|\mathbf{x}) = \frac{\hat{p}_k \cdot \hat{f}_k(\mathbf{x})}{\sum_{\ell=1}^{K} \hat{p}_k \cdot \hat{f}_\ell(\mathbf{x})}.$$

Ces techniques sont particulièrement utiles lorsque les densités de classes sont mal ajustées aux densités paramétriques usuelles. Notons par exemple, que toutes les densités paramétriques sont uni-modales, ce qui n'est bien évidemment pas toujours le cas des densités observées dans des problèmes réels.

Ce chapitre présente un choix de trois méthodes non paramétriques différentes, parmi les nombreuses existantes[1] :

- les fenêtres de Parzen,
- la technique des k plus proches voisins,
- les modèles de mélange.

[1] De nombreuses autres méthodes non paramétriques existent, mais elles sont toutes reliées plus ou moins directement aux trois approches présentées dans ce chapitre.

3.1 Estimation non paramétrique d'un densité

Comment estimer une densité, sans supposer que celle-ci appartient à une famille connue, lorsqu'on dispose de N observations ? L'estimateur le plus simple est sans doute l'histogramme. Dans le cas multi-dimensionnel, le calcul d'un histogramme revient à partitionner l'espace en cellules disjointes de même volume et à estimer la densité d'une cellule comme la proportion d'observations de l'échantillon tombant dans cette cellule. En pratique cette approche n'est pas adaptée car elle produit (avec des échantillons de taille raisonnable) une estimation nulle dans la plupart des cellules. En effet, si les vecteurs forme sont caractérisés par d variables et que chaque variable est partagée en M intervalles, il faut considérer M^d cellules, ce qui représente souvent bien plus que le nombre d'observations disponibles !

Une alternative consiste à utiliser les estimateurs par noyaux. Historiquement introduits par **?**) dans un rapport non publié, ces estimateurs exploitent le fait, que la probabilité qu'un vecteur $\mathbf{X}$ de loi $f()$ tombe dans une région $\mathcal{R}$ avec une probabilité

$$P = \int_{\mathcal{R}} f(\mathbf{y}) d\mathbf{y}.$$

Ainsi, si l'on dispose de la réalisation d'un échantillon i.i.d. $(\mathbf{x}_1, ..., \mathbf{x}_N)$ de loi parente $f()$, alors la probabilité que r de ces vecteurs tombent dans $\mathcal{R}$ suit une loi binomiale $\mathcal{B}(P, N)$ et

$$P(R = r) = C_r^N P^r (1 - P)^{N-r}.$$

L'espérance de la variable aléatoire R est

$$I\!E[R] = N \cdot P,$$

et P est classiquement estimé par $\hat{P} = r/N$. Si le volume V de la région $\mathcal{R}$ est suffisamment petit, alors on peut supposer que $f(\mathbf{x})$ varie peu sur $\mathcal{R}$ et que l'approximation suivante est raisonnable

$$P = \int_{\mathcal{R}} f(\mathbf{y}) d\mathbf{y} \approx V \cdot f(\mathbf{x}),$$

avec $\mathbf{x}$, un vecteur de $\mathcal{R}$. En remplacant P par son estimation, on obtient :

$$\hat{f}(\mathbf{x}) = \frac{r/N}{V}.$$

Il est possible de montrer que sous certaines hypothèses concernant le choix de V et de r, en fonction de N, cet estimateur est convergent (lorsque N tend vers l'infini). Dans le cadre qui nous intéresse, constatons seulement qu'en pratique cet estimateur exige de choisir soit un volume, soit une valeur de r :

- si un volume V est choisi autour de $\mathbf{x}$, alors l'estimation de $\hat{f}_k(\mathbf{x})$ nécessite le comptage du nombre de vecteurs forme, qui appartiennent à ce volume. Cet type de technique est connue sous le nom de fenêtre de Parzen et constitue le sujet de la prochaine section ;

- si un nombre r est fixé, il faut trouver un volume V autour de $\mathbf{x}$ qui contienne r vecteurs forme. Cette méthode est baptisée méthode des k plus proches voisins (KPPV).

3.2 Fenêtres de Parzen

À la suite de **?**), l'approche de Parzen a été proposée par **?**) dans le cas unidimensionnel puis par **?**) dont le nom est resté. La méthode consiste à choisir une fonction fenêtre $\delta(\mathbf{x} - \mathbf{x}_i)$, qui permet d'estimer le rapport r/V autour de $\mathbf{x}$, c'est-à-dire le nombre r par unité de volume :

$$\frac{r}{V} \approx \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i).$$

L'estimateur $\hat{f}()$ de $f()$ s'exprime alors comme :

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i).$$

Pour garantir que $\hat{f}()$ définit bien une densité, il faut que

$$\delta(\mathbf{y}) > 0$$

et

$$\int \delta(\mathbf{y}) d\mathbf{y} = 1.$$

En effet, dans ce cas, ces deux conditions seront aussi satisfaites par $\hat{f}()$. La fonction δ pondère la contribution du vecteur $\mathbf{x}_i$ dans le calcul de la densité au point $\mathbf{x}$. Ainsi, il semble souhaitable, que $\delta(\mathbf{x} - \mathbf{x}_i)$ soit très petite (voir nulle) lorsque $\mathbf{x}$ et $\mathbf{x}_i$ sont très éloignés, et maximum quand $\mathbf{x} = \mathbf{x}_i$. La fonction fenêtre hypercubique et la gaussienne multivariée constituent des exemples de fonction δ.

Notons que la définition de ce type de fonction nécessite toujours de choisir un paramètre h, qui détermine "la zone d'influence" associée à chaque observation $\mathbf{x}_i$. Par exemple, dans le cas de la gaussienne, ce paramètre h est la variance. Si h tend vers 0 alors la fonction δ tend à devenir une Dirac et l'estimateur de la densité est une fonction très chahutée. Si par contre h est choisi très grand par rapport à la distance moyenne séparant les $\mathbf{x}_i$, l'estimateur de la densité sera une fonction très lisse. Ainsi le choix de ce paramètre a une influence déterminante sur l'estimateur $\hat{f}()$. Si h est trop grand alors la densité estimée n'aura aucune précision, et dans le cas contraire elle sera extrêmement variable. Dans le cadre de la discrimination, notons qu'il semble avantageux de considérer des valeurs de h propre à chaque classe. Il y a peu de raisons pour qu'une valeur de h optimale pour une densité de classe donnée, soit optimale pour toutes les autres densités de classe du problème. Une procédure d'estimation

(a) Modélisation par la méthode de noyaux

(b) Modèle gaussien

Figure 3.1: Répartition de la taille d'adultes de sexes masculins modélisée par une méthode non paramétrique (a) et une méthode paramétrique (b) à partir de 10 mesures.

possible de h consiste à utiliser le maximum de vraisemblance par l'intermédiaire d'une validation croisée (chapitre 2).

Dans le contexte de la discrimination, si les densités de classes sont estimées en utilisant une fonction fenêtre δ, les loi *a posteriori* s'expriment :

$$\begin{aligned}\hat{\pi}(k|\mathbf{x}) &= \frac{\hat{p}_k \cdot \hat{f}_k(\mathbf{x})}{\sum_{\ell=1}^{K} \hat{p}_k \cdot \hat{f}_\ell(\mathbf{x})} \\ &= \frac{(\frac{n_k}{N})\frac{1}{n_k}\sum_{i=1}^{n_k}\delta(\mathbf{x}-\mathbf{x}_{i,k})}{\sum_{\ell=1}^{K}(\frac{n_\ell}{N})\frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\delta(\mathbf{x}-\mathbf{x}_{i,\ell})} \\ &= \frac{\sum_{i=1}^{n_k}\delta(\mathbf{x}-\mathbf{x}_{i,k})}{\sum_{i=1}^{N}\delta(\mathbf{x}-\mathbf{x}_i)}\end{aligned}$$

où $\mathbf{x}_{i,k}$ un vecteur forme de la classe k.

La puissance de ce type de méthodes réside dans leur généralité. Elles sont théoriquement à même d'estimer n'importe quel type de densité continue. En pratique, elles sont surtout efficaces, si l'on dispose d'un grand ensemble d'apprentissage. Notons aussi, que des problèmes apparaissent lorsque les vecteurs forme appartiennent à un espace de grande dimension. En effet, dans ce cas la taille de l'ensemble d'apprentissage, nécessaire pour obtenir une certaine précision d'estimation, croît de manière exponentielle avec la dimension de l'espace. De plus, si des fonction δ à support borné sont utilisées, alors le risque d'obtenir des estimations nulles (comme avec les histogrammes). Pour palier ce comportement, une solution consiste à réduire dans un premier temps la dimensionalité des données (chapitre 5).

3.3 Estimation par les k plus proches voisins

Dans la section précédente, nous avons remarqué, qu'un problème majeur de l'approche par fenêtre de Parzen pour estimer une densité $f()$ en $\mathbf{x}$, repose dans le choix du volume considéré autour d'un vecteur forme $\mathbf{x}$, pour N une taille d'échantillon donnée. Une alternative simple consiste à choisir le volume V autour de $\mathbf{x}$, qui contienne exactement r vecteurs forme de l'échantillon observé. Dans ce cas, l'estimateur $\hat{f}()$ de $f()$ s'exprime comme :

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\frac{r}{V_r}.$$

Cette méthode possède l'avantage de choisir le volume V_r en fonction des données, et non plus de manière arbitraire.

L'usage de cette technique pour l'estimation des loi *a posteriori* amène :

$$\begin{aligned}\hat{\pi}(k|\mathbf{x}) &= \frac{\hat{p}_k \cdot \hat{f}_k(\mathbf{x})}{\sum_{\ell=1}^{K} \hat{p}_k \cdot \hat{f}_\ell(\mathbf{x})} \\ &= \frac{\left(\frac{n_k}{N}\right) \cdot \frac{r_k/n_k}{V}}{\frac{r/N}{V}}\end{aligned}$$

$$= \frac{r_k}{r},$$

où r_k est le nombre de vecteurs forme de la classe k parmi les r plus proches voisins de $\mathbf{x}$.

La version la plus simple de cette technique consiste à considérer le plus proche voisin ($r = 1$). D'un point de vue géométrique, la règle du plus proche voisin partage l'espace en un pavage de Voronoï suivant les vecteurs forme de l'ensemble d'apprentissage $\mathcal{F} = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}_N))\}$ et attribue la classe c_i à chaque nouvel individu tombant dans le pavé défini autour $\mathbf{x}_i$.

?) sont parvenus à donner un encadrement de la probabilité d'erreur obtenue par la règle du plus proche voisin, lorsque la taille de l'échantillon tend vers l'infini :

Théorème 3.1 *Si P^* dénote la probabilité d'erreur de Bayes, qui est l'erreur mimimale, et P la probabilité d'erreur limite, lorsque la taille de l'ensemble d'apprentissage tend vers l'infini, de la règle du plus proche voisin alors*

$$P^* \leq P \leq P^*(2 - \frac{K}{K-1}P^*).$$

où K est le nombre de classes.

Dans le cas où r voisins sont pris en compte, la règle de décision consiste à déterminer la classe majoritaire parmi ces r voisins. Par rapport aux fenêtres de Parzen, cette approche possède l'avantage de produire des estimateurs des lois *a posteriori* qui ne sont jamais nuls, même loin de tout vecteur forme de l'ensemble d'apprentissage.

La règle des r plus proches voisins peut être étendue de manière prendre en compte le doute (**?**) : il suffit de considérer r plus proches voisins et de décider de douter si la classe majoritaire représente moins de s voisins parmi les r pris en compte.

Remarquons que les méthodes de fenêtres de Parzen et des r plus proches voisins requierent le stockage de tout l'ensemble d'apprentissage. Elles ne nécessitent pas d'apprentissage mais beaucoup de calculs pour prendre une décision.

3.4 Modèle de mélange et algorithme EM

Les modèles de mélanges se situent à la frontière entre les modèles paramétriques et non paramétriques. L'estimation des paramètres de ces modèles est une tâche compliquée et l'une des méthodes d'estimation les plus populaire, dans ce contexte, est l'algorithme EM.

3.4.1 Modèle de mélange gaussien

En 1894, Karl Pearson publiait un article sur l'estimation par la méthode des moments des cinq paramètres d'une densité mélange de deux distribution normales univariées

(**?**). Depuis, ce genre de modèle connaît un certain succès et a été à l'origine de nombreuses applications.

D'une manière très générale, les mélanges de densité sont des distributions de probabilité de la forme suivante :

$$f(\mathbf{x}) = \int h(\theta) \cdot f(\mathbf{x}|\theta) d\theta \tag{3.1}$$

où $f(\mathbf{x}|\theta)$ est une densité paramétrique conditionnelle définie par le paramètre θ et $h(\theta)$ est la densité de mélange.

Lorsque la densité de probabilité $h(\theta)$ est discrète et prend ses valeurs sur un ensemble fini $(\theta_1, \cdots, \theta_K)$ avec les probabilités $(p_1, \cdots, p_K)$ (avec $\sum_{k=1}^{K} p_k = 1$), la densité f s'écrit

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}|\theta_k), \tag{3.2}$$

et on parle de mélange fini. Ce genre de densité apparaît naturellement lorsque la population considérée est formée de plusieurs sous-populations qui ont des densités différentes.

Exemple 3.1 En mécanique, l'étude d'un matériau passe souvent par une phase pratique d'essais de traction. On tire sur le matériau pour observer la déformation et temps de rupture. En pratique, une distribution de Weibull est souvent un bon modèle statistique du temps de rupture. Comme le matériau peut se rompre pour diverses raisons, un mélange de distributions de Weibull permet de modéliser le phénomène. Dans ce cas, il y aura autant de composants que de raisons de rupture.

△

Notons aussi que les mélanges finis peuvent modéliser des distributions de probabilités "biscornues" dont les modes ne correspondent pas forcement à la présence d'une sous-population. Dans le cadre de la discrimination, c'est cette dernière propriétés qui est exploitée. Si les densités de classes sont estimés par des mélange de loi,

$$\hat{f}_k(\mathbf{x}) = \sum_{i=1}^{m_k} p_{ki} \hat{f}_{ki}(\mathbf{x}),$$

la loi *a posteriori* s'exprime alors comme :

$$\hat{\pi}(k|\mathbf{x}) \quad = \quad \frac{\hat{p}_k \cdot \sum_{i=1}^{m_k} p_{ki} \hat{f}_{ki}(\mathbf{x})}{\sum_{\ell=1}^{K} \hat{p}_k \cdot \sum_{i=1}^{m_\ell} p_{\ell i} \hat{f}_{\ell i}(\mathbf{x})}.$$

Les mélanges sont un intermédiaire, un compromis entre approche paramétrique et non paramétrique[2]. Dans ce document, nous nous concentrerons sur le modèle de mélange gaussien (Figure 3.2), qui est de loin le plus populaire.

[2]Dans un souci de simplification des notations, tout le reste du chapitre utilise les conventions

(a) Trois distributions gaussiennes (b) Mélange de distributions

Figure 3.2: Exemple d'un mélange gaussien

Le problème consiste à estimer les paramètres du mélange. Avant de résoudre ce genre de problème, il faut s'assurer qu'il est bien posé, c'est-à-dire qu'il admet une solution unique et donc que les composants du mélange sont effectivement identifiables.

Exemple 3.2 Le mélange de deux lois uniformes n'est pas identifiable. Prenons par exemple les deux distributions suivantes :

$$f(x) = \frac{1}{3}U[-1,1] + \frac{2}{3}U[-2,2]$$
$$f(x) = \frac{1}{2}U[-2,1] + \frac{1}{2}U[-1,2]$$

Elles sont identiques et il existe même une infinité de mélanges de lois uniformes qui sont identiques aux deux densités précédentes.

△

On peut montrer que les mélanges gaussiens (ainsi que les mélanges exponentiels, de Poisson et de Cauchy) sont identifiables. Dans la suite du document, nous détaillerons plusieurs méthodes d'estimation adaptées à ce genre de modèles de mélange identifiable.

3.4.2 L'algorithme EM

Le principe d'information manquante

Dans certains problèmes, l'échantillon de données disponible ne permet pas de calculer facilement les estimateurs du maximum de vraisemblance. C'est par exemple le cas pour l'estimation des paramètres d'un mélange fini de densités de probabilité.

Exemple 3.3 (**?**) La taille d'un flétan (poisson de la mer baltique) d'un âge donné est distribuée suivant un mélange de deux lois gaussiennes correspondant aux deux distributions relatives aux mâles et femelles :

$$f(\mathbf{x}|\Phi) = p_1 f_1(\mathbf{x}|\mu_1, \sigma_1) + p_2 f_2(\mathbf{x}|\mu_2, \sigma_2) \tag{3.3}$$

où les p_k sont les proportions du mélange ($0 < p_k < 1$, pour $k = 1, 2$ et $\sum_k p_k = 1$), $f_k(\mathbf{x}|\mu_k, \sigma_k)$ est une loi de Gauss de moyenne μ_k et d'écart type σ_k, et $\Phi = (p_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$. L'estimation de Φ est un problème simple si les mesures prises spécifient la taille et le sexe de chaque poisson considéré. Malheureusement le sexe du flétan est difficile à déterminer et il faut estimer le vecteur Φ à l'aide de données incomplètes.

---

usuelles suivantes :

- $f(\mathbf{x}|\Phi)$ sera utilisée à la place de $\hat{f}_k(x)$ et dénotera une densité mélange, qui dans notre cas est utilisée comme densité de classe ;
- $f_k(\mathbf{x})$ sera utilisée à la place de $f_{ik}(\mathbf{x})$ et dénotera une composante du mélange.

△

Pour maximiser la vraisemblance de ce type de données, qualifiées de données incomplètes, il est souvent avantageux de poser le problème pour un jeu hypothétique de données complètes. Cette façon d'aborder le problème conduit à la formulation d'un algorithme itératif qui permet de calculer des estimateurs des paramètres inconnus. Dempster, Laird et Rubin (1977) ont baptisé cet algorithme, basé sur le principe de l'information manquante, algorithme EM (Expectation Maximization), et ont donné des nombreux exemples de son application à des problèmes aussi variés que le calcul des estimateurs du m.v. des paramètres d'une loi multinomiale, d'un mélange fini de densités ou l'estimation d'hyperparamètres dans un cadre bayésien.

D'une manière très générale deux espaces mesurables sont considérés : $\mathcal{X}$, l'espace des données observées (ou données incomplètes) et $\mathcal{Y}$ l'espace des données complètes. Soient deux vecteurs $\mathbf{x} \in \mathcal{X}$ et $\mathbf{y} \in \mathcal{Y}$, de densité respective $f(\mathbf{x}|\Phi)$ et $g(\mathbf{y}|\Phi)$.

Le but de l'algorithme est de calculer l'estimateur du m.v. du vecteur de paramètres inconnus, Φ, en utilisant les relations qui existent entre $\mathbf{x}$ et $\mathbf{y}$.

En pratique $\mathbf{y}$ n'est pas observé et contient des données manquantes, des paramètres inconnus, des données inobservables (e.g. le sexe des flétans dans l'exemple précédent).

On note $k(\mathbf{y}|\mathbf{x}, \Phi)$ la densité conditionnelle des données complètes connaissant les données observées :

$$f(\mathbf{x}|\Phi) = \frac{g(\mathbf{y}|\Phi)}{k(\mathbf{y}|\mathbf{x}, \Phi)}. \tag{3.4}$$

En prenant le logarithme, on obtient :

$$L(\Phi; \mathbf{x}) = L(\Phi; \mathbf{y}) - L(\Phi; \mathbf{y}|\mathbf{x}), \tag{3.5}$$

où $L(\Phi; \mathbf{y})$ et $L(\Phi; \mathbf{x})$ sont les log-vraisemblances de Φ en considérant respectivement les données complètes et les données observées. De même $L(\Phi; \mathbf{y}|\mathbf{x})$ représente la log-vraisemblance de Φ tenant compte de la densité conditionnelle de $\mathbf{y}$ sachant $\mathbf{x}$.

Considérons Φ_d une valeur donnée du vecteur Φ. En prenant de chaque coté de l'équation 3.5, l'espérance pour la loi $k(\mathbf{y}|\mathbf{x}, \Phi_d)$, on peut écrire :

$$L(\Phi; \mathbf{x}) = Q(\Phi|\Phi_d) - H(\Phi|\Phi_d), \tag{3.6}$$

où

$$Q(\Phi|\Phi_d) = I\!E^k[L(\Phi; \mathbf{y})|\mathbf{x}, \Phi_d];$$
$$H(\Phi|\Phi_d) = I\!E^k[L(\Phi; \mathbf{y}|\mathbf{x})|\mathbf{x}, \Phi_d].$$

Notons que l'inégalité de Jensen (Dempster *et al.* 1977) permet de montrer que la valeur de Φ, qui maximise $H(\Phi|\Phi_d)$, est Φ_d. La valeur Φ^+ de Φ qui maximise $Q(\Phi|\Phi_d)$ est une fonction de Φ_d :

$$\Phi^+ = M(\Phi_d). \tag{3.7}$$

Soit Φ^* le maximum de vraisemblance cherché. Si l'on pose

$$\Phi_d = \Phi^*$$

il est alors évident que la valeur Φ^* maximise

$$L(\Phi; \mathbf{x}) + H(\Phi|\Phi^*).$$

De cette constatation, on déduit que Φ^* maximise $Q(\Phi|\Phi^*)$. Ainsi, Φ^* est un point fixe de la fonction $M(\Phi)$, et ceci suggère un algorithme itératif de type point fixe qui calcule le paramètre Φ^{q+1} à partir d'une valeur Φ^q :

- **Etape d'Estimation :** Déterminer $Q(\Phi|\Phi^q) = I\!E^k[L(\Phi; \mathbf{y})|\mathbf{x}, \Phi^q]$
- **Etape de Maximisation :** Calculer $\Phi^{q+1} = M(\Phi^q)$. Φ^{q+1} vérifie alors

$$\Phi^{q+1} = arg \max_{\Phi} Q(\Phi|\Phi^q)$$

La propriété fondamentale de l'algorithme EM est que chaque itération augmente la vraisemblance des paramètres à estimer. En effet, suite à l'étape de maximisation on a

$$Q(\Phi^{q+1}|\Phi^q) \geq Q(\Phi^q|\Phi^q)$$

et d'après l'inégalité de Jensen (Dempster *et al.* 1977) :

$$H(\Phi^{q+1}|\Phi^q) \leq H(\Phi^q|\Phi^q),$$

donc

$$L(\Phi^{q+1}; \mathbf{x}) \geq L(\Phi^q; \mathbf{x}).$$

Dans un cadre général la convergence de l'algorithme n'est pas démontrée (la démonstration de Dempster, Laird et Rubin en 1977 était fausse) et si l'algorithme converge vers un point fixe, on est seulement sûr que c'est un point stationnaire de la vraisemblance et pas obligatoirement un maximum local, mais dans le cadre de l'estimation des paramètres d'un mélange fini, qui nous intéresse particulièrement, **?**) ont démontré le théorème de convergence locale suivant :

Théorème 3.2 *(**?**) Soit un mélange de densités exponentielles, supposons que $I(\Phi)$, la matrice d'information de Fisher associée aux paramètres du mélange est définie positive pour Φ^* les vraies valeurs des paramètres, si les proportions sont positives, alors pour n suffisamment grand, l'unique solution presque sûrement consistante Φ_n des équations de vraisemblance existe presque sûrement, et la suite $\{\Phi^q\}$ des itérés de l'algorithme EM converge vers Φ_n pourvu que la position initiale Φ^0 soit suffisamment proche de Φ_n; de plus il existe une norme sur l'espace des paramètres pour laquelle il existe λ, $0 \leq \lambda < 1$, pour laquelle :*

$$\|\Phi^{q+1} - \Phi_n\| \leq \lambda \|\Phi^q - \Phi_n\|, \forall q \geq 0.$$

D'après ce théorème, et avec un peu de pratique, on s'aperçoit que l'initialisation de l'algorithme conditionne la qualité du résultat. Si la position initiale choisie est très "éloignée" de la vraie valeur des paramètres, l'algorithme EM risque de converger vers une solution singulière.

L'algorithme EM converge linéairement et dans certaines situations peut s'avérer particulièrement lent. Ainsi, lorsque les composants du mélange sont mal séparés, le coefficient λ sera proche de 1 et un grand nombre d'itérations sera nécessaire à la convergence.

Pour pallier ce problème de vitesse de convergence, Redner et Walker (1984) ont suggéré l'utilisation de méthodes d'optimisation qui ont une meilleure vitesse de convergence comme celle de Newton. La méthode de Newton est itérative. A partir d'une position initiale Φ^0, une suite d'itérés est calculée comme suit :

$$\Phi^{q+1} = \Phi^q - H(\Phi^q)^{-1}\nabla_\Phi L(\Phi^q; \mathbf{x}), \tag{3.8}$$

où $H(\Phi^q)$ est la matrice hessienne de $L(\Phi_q; \mathbf{x})$. Cette méthode a une vitesse de convergence quadratique ; c'est-à-dire qu'il existe une constante λ, telle que :

$$\|\Phi^{q+1} - \Phi_n\| \leq \lambda \|\Phi^q - \Phi_n\|^2.$$

La convergence quadratique est beaucoup plus rapide que la convergence linéaire mais le calcul de l'inverse de la matrice hessienne est très coûteux. Une autre méthode possible est celle de quasi Newton, qui approxime la matrice hessienne et réduit ainsi la complexité algorithmique de la méthode de Newton tout en ayant une convergence supra-linéaire, donc supérieure à celle de l'algorithme EM.

Malgré les qualités des méthodes de Newton, l'algorithme EM reste très utilisé pour plusieurs raisons. En effet, chaque itération nécessite peu de calculs et même dans les cas où la convergence vers les vraies valeurs des paramètres est lente, la convergence de la vraisemblance reste très rapide (**?**). Ainsi les premières itérations produisent des bonnes valeurs des paramètres et les nombreuses autres augmentent peu la vraisemblance. **?**) remarquent :

> In the context of the current litterature on learning, in which the predictive aspects of data modeling is emphasized at the expense of the traditonal Fisherian statistician's concern over the "true" value of the parameters, such rapid convergence in likelihood is a major desideratum of a learning algorithm and undercuts the critique of EM as a "slow" algorithm.

Lorsque la convergence de l'algorithme EM est lente (composantes du mélange mal séparées), les matrices Hessiennes sont mal conditionnées et les méthodes super-linéaires et de quadratiques ont aussi des problèmes. De plus dans le cas des modèles de mélange gaussien, l'algorithme EM peut être considéré comme une montée de gradient projeté (Xu et Jordan 1995) ; les deux étapes de l'algorithme se résument à l'équation suivante :

$$\Phi^{q+1} = \Phi^q - P(\Phi^q)\nabla_\Phi L(\Phi^q; \mathbf{x}) \tag{3.9}$$

où $P(\Phi^q)$ est une matrice de projection calculée à chaque itération. Il est alors possible de montrer que sous certaines conditions l'algorithme EM approxime une méthode superlinéaire.

Application au modèle de mélange

Pour toutes les raisons mentionnées précédemment, l'algorithme EM est très utilisé pour l'estimation des paramètres d'un modèle de mélange de densité de probabilité. Dans ce contexte, son utilisation est bien antérieure à l'article de Dempster, Laird et Rubin (1977) : **?**) proposait déjà un algorithme identique pour identifier les paramètres d'un mélange de deux gaussiennes multidimensionnelles. **?**), de manière indépendante, décrivait un algorithme de classification automatique probabiliste destiné à l'estimation des paramètres de mélanges de K lois de Bernoulli, ou de Gauss multivariées. Cet article remarquable introduisait ainsi l'algorithme EM pour obtenir une partition floue, alors que la notion de flou en classification ne se développa qu'à partir de 1974 principalement sous l'impulsion de **?**).

Dans le cadre d'un modèle de mélange, le problème d'estimation des paramètres se pose comme suit : on dispose d'un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_N)$ d'une variable aléatoire à valeurs dans $\mathbb{R}^d$ de densité :

$$f(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{3.10}$$

où les p_k sont les proportions du mélange ($0 < p_k < 1$, pour $k = 1, .., K$ et $\sum_k p_k = 1$) et $f_k(\mathbf{x}|\theta_k)$ est une loi complètement déterminée par la connaissance du vecteur θ_k.

Posons le problème de l'estimation de $\Phi = (p_1, ..., p_k, \theta_1, ..., \theta_K)$ sous une forme traitable par le principe d'information manquante. Considérons que l'échantillon observé $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ est incomplet. L'échantillon complet s'écrit $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ avec $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{z}_i)$. $\mathbf{z}_i = (z_{ik}, k = 1, ..., K)$ est un vecteur qui indique de quelle composante du mélange est issu $\mathbf{x}_i$ ($z_{ik} \in \{0, 1\}$ et $\sum_{k=1}^{K} z_{ik} = 1$) : $z_{ik} = 1$ signifie que $\mathbf{x}_i$ provient de la $k$composante. Indiçons les paramètres à estimer par $\mathbf{z}_i$ à la place de k lorsque $z_{ik} = 1$ et écrivons les densités des deux échantillons $\mathbf{x}$ et $\mathbf{y}$:

$$f(\mathbf{x}|\Phi) = \prod_{i=1}^{N} f(\mathbf{x}_i|\Phi) = \prod_{i=1}^{N} \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{3.11}$$

et

$$g(\mathbf{y}|\Phi) = \prod_{i=1}^{N} p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i}). \tag{3.12}$$

$k(\mathbf{y}|\mathbf{x}, \Phi)$ la densité conditionnelle des données complètes connaissant les données observées s'exprime par :

$$k(\mathbf{y}|\mathbf{x}, \Phi) = \prod_{i=1}^{N} k(\mathbf{y}_i|\mathbf{x}_i; \Phi) = \prod_{i=1}^{N} \frac{p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i})}{\sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k)}. \tag{3.13}$$

Ainsi dans le cas particulier des modèles de mélanges les quantités Q et H de l'équation 3.6 deviennent :

$$\begin{aligned}
Q(\Phi|\Phi^q) &= I\!E^k[L(\Phi;\mathbf{y})|\mathbf{x},\Phi^q] = I\!E^k[\log\prod_{i=1}^{N} p_{\mathbf{Z}_i} f_{\mathbf{Z}_i}(\mathbf{x}_i|\theta_{\mathbf{Z}_i})|\mathbf{x},\Phi^q] \\
&= \sum_{i=1}^{N} I\!E^k[\log p_{\mathbf{Z}_i} f_{\mathbf{Z}_i}(\mathbf{x}_i|\theta_{\mathbf{Z}_i})|\mathbf{x},\Phi^q] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log p_k f_k(\mathbf{x}_i|\theta k) \\
H(\Phi|\Phi^q) &= I\!E^k[L(\Phi;\mathbf{y}|\mathbf{x})|\mathbf{x},\Phi_d] = I\!E^k[\log\prod_{i=1}^{N} k(\mathbf{y}_i|\mathbf{x}_i;\Phi)|\mathbf{x},\Phi^q] \\
&= \sum_{i=1}^{N} I\!E^k[\log k(\mathbf{y}_i|\mathbf{x}_i;\Phi)|\mathbf{x},\Phi^q] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log t_k(\mathbf{x}_i)
\end{aligned}$$

avec

$$t_k(\mathbf{x}_i)^q = \frac{p_k^q f_k(\mathbf{x}_i|\theta_k^q)}{f(\mathbf{x}_i)}. \tag{3.14}$$

Dans la suite de cette section, nous considérons uniquement le cas des mélanges gaussiens qui sont de loin les plus utilisés en classification automatique. Le mélange est alors paramétré par le vecteur $\Phi^q = (p_1^q, ..., p_{K-1}^q, \theta_1^q, ..., \theta_K^q)$ où $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ et les deux étapes de l'algorithme EM s'écrivent :

- **Etape E :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ en utilisant $\Phi^q = (p_1^q, ..., p_k^q, \theta_1^q, ..., \theta_K^q)$.
- **Etape M :** Calcul de Φ^{q+1} qui maximise

$$Q(\Phi|\Phi^q) = \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log p_k f_k(\mathbf{x}_i|\theta_k).$$

Les estimateurs du maximum de vraisemblance s'écrivent alors :

$$\boldsymbol{\mu}_k^{q+1} = \frac{\sum_{i=1}^{N} t_k(\mathbf{x}_i)^q \cdot \mathbf{x}_i}{n_k^q}; \tag{3.15}$$

$$\boldsymbol{\Sigma}_k^{q+1} = \frac{1}{n_k}\sum_{i=1}^{N} t_k(\mathbf{x}_i)^q (\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})(\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})^t; \tag{3.16}$$

$$p_k^{q+1} = \frac{n_k^q}{N}, \tag{3.17}$$

où $n_k^q = \sum_{i=1}^{N} t_k(\mathbf{x}_i)^q$.

A la convergence, l'algorithme EM fournit une estimation des paramètres du mélange, ainsi que des probabilités $k((\mathbf{x}_i, \mathbf{z}_i)|\mathbf{x}_i)$.

Chapter 4

Fonctions discriminantes et méthodes géométriques

Dans les chapitres précédents, les fonctions discriminantes étaient obtenues par l'intermédiaire des densités de classes. Une autre alternative, baptisée discrimination logistique (**?**), et non développée dans ces notes, consiste à estimer directement les paramètres des lois *a posteriori* qui sont alors utilisées comme fonctions $d_k()$. Dans ce chapitre, nous présentons deux approches "non statistique" qui permettent de définir des fonctions discriminantes :

- les méthodes géométriques,
- les réseaux de neurones à couches.

4.1 Approche géométrique et métriques de discrimination

Les méthodes géométriques considèrent les N vecteurs forme de l'ensemble d'apprentissage comme un nuage de $\mathbb{R}^d$ partagé en K sous nuages de centres de gravité $\mathbf{m}_1, \cdots, \mathbf{m}_K$.

L'idée de base consiste à définir les fonctions discriminantes à partir de distances dans l'espace des paramètres. Cela revient à spécifier une fonction de similitude entre un nouvel individu, qui est considéré comme un point dans $\mathbb{R}^d$, et un sous nuage donné. Le nouvel individu est affecté à la classe avec laquelle il est le plus similaire. Les questions suivantes emmèrgent :

- Comment combiner les distances entre les individus d'un nuage donné à un nouvel individu pour obtenir une fonction de similitude (fonction discriminante) ?

- Quelle type de distance prendre en compte ?

Commençons par envisager la deuxième question. Si la distance euclidienne est utilisée, cela revient à accorder la même importance à chacune des variables. Il semble pourtant raisonnable de penser que toutes les variables ne possèdent pas le même pouvoir discriminant et qu'il est plus judicieux de travailler sur des variables transformées. Si l'on considère l'ensemble des transformations linéaires possibles sur les variables, alors cela revient à se limiter aux métriques définies par une forme quadratique

$$\delta^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^t M (\mathbf{x} - \mathbf{y}).$$

En pratique, l'approche géométrique repose donc sur le choix *a priori* des fonctions discriminantes, combinaison de distances, et sur la recherche d'une métrique optimale au sens d'un certain critère.

4.1.1 La règle de Mahalonobis Fisher

Une approche populaire consiste à définir les fonctions discriminantes en utilisant les distances aux centres de gravité des classes. Ainsi le vecteur forme $\mathbf{x}$ est affecté à la classe k si

$$k = arg \min_{\ell} (\mathbf{x} - \mathbf{m}_\ell)^t M (\mathbf{x} - \mathbf{m}_\ell)$$

La métrique M peut par exemple être choisie de manière à ce que les individus composant les classes soient le moins dispersés possible autour de leurs centres de gravité $\mathbf{m}_k$. Traduit en terme de critère, cette dernière exigence peut s'exprimer comme la minimisation d'une inertie $I(M)$:

$$\begin{cases} M = arg \min_Q \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{1}{N} (\mathbf{x}_{i,k} - \mathbf{m}_k)^t Q (\mathbf{x}_{i,k} - \mathbf{m}_k), \\ |M| = 1 \end{cases}$$

Cette expression peut se mettre sous la forme :

$$\begin{cases} M = arg \min_Q trace[WQ], \\ |M| = 1 \end{cases}$$

avec $W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{1}{N} (\mathbf{x}_{i,k} - \mathbf{m}_k)(\mathbf{x}_{i,k} - \mathbf{m}_k)^t$. Notons que W est un estimateur de la matrice de covariance supposée commune à toutes les classes. Supposons que la matrice W est inversible (ce qui est très souvent le cas). Comme W et M sont symétriques définies positives, les valeurs propres $\lambda_1, \cdots, \lambda_d$ de la matrices WM sont toutes strictement positives. Remarquons que $trace[WM] = \sum_{i=1}^{d} \lambda_i$ et $|WM| = \prod_{i=1}^{d} \lambda_i$. Comme $|M| = 1$, on en déduit que le produit des valeurs propres est constant et égal au déterminant de W (car $|WM| = |W| \cdot |M|$). Le problème de minimisation prend alors la forme suivante :

$$\begin{cases} \min \sum_{i=1}^{d} \lambda_i, \\ \prod_{i=1}^{d} \lambda_i = |W|. \end{cases}$$

En écrivant le lagrangien, un calcul de gradient donne

$$\lambda_1 = \cdots = \lambda_d = |W|^{1/d} = \lambda.$$

La décomposition de la matrice WM sur sa base de vecteurs propres U amène

$$\begin{aligned} WM &= U^t\lambda \cdot IU, \\ &= \lambda \cdot I. \end{aligned}$$

d'où l'on déduit que $M = \lambda \cdot W^{-1}$.

La règle de décision obtenue est la règle de Mahalonobis Fisher : un vecteur forme $\mathbf{x}$ est affecté à la classe la plus proche au sens de la distance :

$$\delta(\mathbf{x}, \mathbf{m}_k) = [(\mathbf{x} - \boldsymbol{\mu}_k)^t W^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)]^{\frac{1}{2}}.$$

Comme nous l'avons déjà mentionné dans le second chapitre, cette règle de décision est linéaire et sépare les classes voisines par des hyperplans.

4.1.2 L'approche de Sebestyen

Un autre exemple d'approche géométrique a été publié par Sebestyen en 1962 (**?**). La fonction discriminante proposé est basée sur la somme des distances de l'individu $\mathbf{x}$ à classer à tous les individus d'un sous-nuage donné. Ainsi $\mathbf{x}$ sera affecté à la classe k si :

$$k = arg\min_{\ell} \frac{1}{n_k} \sum_{i=1}^{n_k} (\mathbf{x} - \mathbf{x}_{i,k})^t M_k (\mathbf{x} - \mathbf{x}_{i,k}).$$

Une métrique M_k, par classe, est choisie de manière à minimiser la distance moyenne entre les individus d'un sous nuage (groupe, classe) :

$$\begin{cases} M_k = arg\min_Q \frac{1}{n_k(n_k-1)} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} (\mathbf{x}_{i,k} - \mathbf{x}_{j,k})^t Q (\mathbf{x}_{i,k} - \mathbf{x}_{j,k}), \\ |M_k| = 1 \end{cases}$$

La règle de décision obtenue affecte $\mathbf{x}$ à la classe k, si :

$$k = arg\min_{\ell} (\mathbf{x} - \mathbf{m}_\ell)^t \boldsymbol{\Sigma}_\ell^{-1} (\mathbf{x} - \mathbf{m}_\ell),$$

où $\boldsymbol{\Sigma}_k$ est la matrice de covariance estimée de la classe k.

Remarquons que cette règle est exactement la même que celle obtenue dans le cas gaussien sous hypothèse d' hétéroscédasticité, lorsque les proportions des classes sont toutes égales et que les matrices de covariance ont même déterminant (c'est à dire occupent un même volume dans l'espace).

4.2 Réseaux de neurones à couches

Historiquement, la première modélisation du neurone a été suggérée dans les années quarante par Mac Culloch et Pitts (**?**). Au début des années soixante, **?**) présentait un modèle très simple de réseau de neurones inspiré du système visuel : le perceptron. Ce modèle suscita beaucoup d'enthousiasme, jusqu'à la publication d'un livre de **?**), qui démontra les limites du modèles. La renaissance des réseaux de neurones (de type perceptron) est à attribuer aux idées novatrices de **?**).

Il existe une grande variété de réseaux de neurones. Cette section est uniquement consacrée aux réseaux multicouches. D'un point de vue mathématique, un réseau de neurones à couches, est une fonction très flexible () de $I\!R^d$ dans $I\!R^K$, qui est généralement définie par de nombreux paramètres $\mathbf{w} = \{w_{ij}\}$. Un réseau de neurones peut être utilisé pour faire de la régression non linéaire, mais aussi pour définir un classifieur (qui est bien une fonction de $I\!R^d$ dans $I\!R^K$).

4.2.1 Des origines

Le neurone formel proposé par Mac Culloch et Pitts est une unité qui en fonction de la somme pondérée de signaux d'entrée transmet une réponse binaire. C'est une fonction seuil de $I\!R^d$ dans $0, 1$ de la forme :

$$y_j(t) = I\!I_{[\sum_i w_{ij} \cdot y_i(t-1) > b_j]}$$

où

- $y_j(t)$ est la sortie du neurone j au temps t. D'un point de vue biologique, c'est la valeur transmise par l'axone du neurone ;
- w_{ij} est le poids de la connexion qui va du neurone i vers le neurone j. Cette valeur est donc une caractéristique de la dendrite qui transmet le signal vers le neurone j.
- les $y_i(t)$ sont les signaux d'entrée (qui peuvent provenir d'autre neurones) au temps t qui sont transmis par l'intermédiaire des dendrites.
- b_j, est le seuil au delà duquel le neurone sera activé.

Ce modèle, très simplifié de neurone, est la la base du perceptron de **?**). Avant de présenter sous un "angle connexioniste" le perceptron, qui possède essentiellement un intérêt historique, commençons par une digression sur les fonctions discriminantes linéaires. L'approche linéaire du problème de classement, dans le cadre des fonctions discriminantes considère que celles ci sont des combinaisons linéaires des vecteurs formes à classer :

$$d_k(\mathbf{x}) = \mathbf{w}^t \cdot \mathbf{x} + w_0.$$

Si l'on se limite à deux classes, deux fonctions discriminantes sont à définir, en fonction de l'ensemble d'apprentissage $\mathcal{F}$:

$$\begin{cases} d_1(\mathbf{x}) = \mathbf{w}_1^t \cdot \mathbf{x} + w_{01} \\ d_2(\mathbf{x}) = \mathbf{w}_2^t \cdot \mathbf{x} + w_{02} \end{cases}$$

Un vecteur forme est classé dans la première classe si :

$$\begin{aligned} d_1(\mathbf{x}) &> d_2(\mathbf{x}), \\ (\mathbf{w}_1^t - \mathbf{w}_2^t)\mathbf{x} + (w_{01} - w_{02}) &> 0. \end{aligned}$$

Une astuce d'écriture couramment utilisée consiste à changer la dimension et le signe des vecteurs formes :

$$\mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \text{ si } \mathbf{x} \in \mathcal{C}_1 \text{ et, } \mathbf{y} = \begin{bmatrix} -1 \\ -\mathbf{x} \end{bmatrix} \text{ sinon}$$

Si l'on note,

$$\mathbf{a} = \begin{bmatrix} w_{01} - w_{02} \\ \mathbf{w}_1 - \mathbf{w}_2 \end{bmatrix},$$

alors on peut dire qu'un vecteur forme $\mathbf{x}$ de l'ensemble d'apprentissage est bien classé si :

$$\mathbf{a}^t\mathbf{y} > 0.$$

Comment définir les composantes du vecteur $\mathbf{a}$ pour obtenir un bon classifieur ? Une approche naturelle consiste à :

- poser un critère qui définisse formellement, ce qu'est un "bon" vecteur $\mathbf{a}$;
- choisir une méthode d'optimisation qui permette de trouver un optimum (souvent local du critère).

Le critère le plus évident est bien sûr le nombre de vecteurs forme mal classé de l'ensemble d'apprentissage. Malheureusement, ce critère n'est pas continu par rapport au vecteur $\mathbf{a}$ et pose des problème d'optimisation. De nombreux autres critères, qui ont de meilleures propriétés, ont été proposés. Citons par exemple le critère des moindres carrés :

$$E_s(\mathbf{a}) = \sum_{\mathbf{y}:\ \mathbf{a}^t\mathbf{y} \leq b} (\mathbf{a}^t\mathbf{y} - b)^2$$

où b est un seuil permettant d'éviter la solution $\mathbf{a} = 0$, qui ne possède aucun sens pour le problème de classement. Notons que la somme est effectuée uniquement sur les vecteurs mal classés ($\mathbf{y} :\ \mathbf{a}^t\mathbf{y} \leq b$).

Le critère du perceptron s'exprime comme :

$$E_p(\mathbf{a}) = \sum_{\mathbf{y}:\ \mathbf{a}^t\mathbf{y} \leq b} (b - \mathbf{a}^t\mathbf{y}).$$

La minimisation de ce critère peut se faire par une simple descente de gradient :

$$\begin{aligned} \mathbf{a}_{k+1} &= \mathbf{a}_k - \rho_k \nabla^t E_p(\mathbf{a}_k) \\ &= \mathbf{a}_k + \rho_k \sum_{\mathbf{y}:\ \mathbf{a}^t\mathbf{y} \leq b} \mathbf{y} \end{aligned}$$

où ρ_k est le pas à l'étape k. Dans la terminologie connexioniste, ce type d'algorithme est dit "batch" car une seule modification des paramètres prend en compte tous les vecteurs forme de l'ensemble d'apprentissage.

L'algorithme initial proposé par Rosenblatt minimise le critère pour la présentation d'un seul vecteur forme à la fois et prend la forme :

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \rho_k \mathbb{1}_{[\mathbf{y}:\ \mathbf{a}^t\mathbf{y} \leq b]} \mathbf{y}$$

Intuitivement, cette règle se comprend facilement : le vecteur poids $\mathbf{a}$ est modifié seulement lorsqu'un vecteur forme mal classé est présenté. Cette dernière forme d'optimisation, parfois qualifiée de "on-line", est très courante dans le domaine des réseaux de neurones, et se justifie dans le cadre de la théorie de l'approximation stochastique. **?**) cite trois raisons qui motivent l'utilisation d'algorithmes "on-line" :

- d'un point de vue biologique, il semble plus "naturel" d'apprendre un peu à chaque expérience nouvelle ;
- d'un point calculatoire, ce type d'algorithme peut converger plus rapidement qu'une version "batch" ;
- enfin, l'introduction de bruit, par l'intermédiaire du choix aléatoire des vecteurs forme, évite peut être de tomber dans des mimima locaux.

Mais quel est le rapport entre les fonctions discriminantes linéaires et les réseaux de neurones ? Si l'on se limite (comme dans la discussion précédente) à la considération de deux classes, une fonction discriminante linéaire peut se mettre sous la forme d'un réseau comportant un seul neurone à seuil binaire comme le montre la figure 4.1. Le calcul du vecteur $\mathbf{a}$ optimal peut s'interpréter comme un "apprentissage" à discriminer entre les vecteurs forme de deux classes distinctes. À la présentation d'un vecteur forme $\mathbf{x}$, le neurone répond 1 si le vecteur forme est classé dans la première classe et 0 sinon.

Les limitations du perceptron résident essentiellement dans son caractère de séparateur linéaire. (**?**) ont proposé un type de réseau plus complexe dépassant cette limitation : les réseaux de neurones à couches.

4.2.2 Réseaux à couches

L'idée à la base des réseaux multicouches est l'utilisation d'une fonction d'activation dérivable pour modéliser le neurone. Les fonctions principalement utilisées sont :

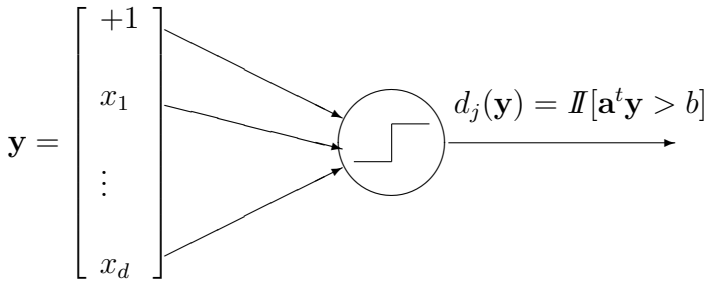

Figure 4.1: Représentation neuronale d'une fonction discriminante linéaire

- la fonction linéaire $d_j(x_i) = a \cdot x_j$,
- la fonction logistique $d_j(x_j) = e^{x_j}/(1 + e^{x_j})$,
- et la tangente hyperbolique $d_j(x_j) = tanh(x_j)$.

où x_j est la somme pondérée des entrées y_i (à ne pas confondre avec $\mathbf{x}$ un vecteur forme) :

$$x_j = \sum_{i \to j} w_{ij} \cdot y_i.$$

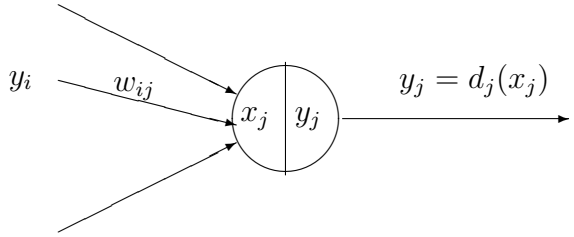

Figure 4.2: Un neurone à fonction d'activation dérivable

Un réseau multicouches est défini comme un ensemble de neurones qui peuvent être numérotés de manière à ce que les connexions aillent toujours d'un neurone i vers un neurone j avec $i < j$. En pratique les neurones sont regroupés en couches. Les neurones d'une couche donnée communiquent seulement avec les couches d'ordre supérieur. Le réseau comporte

- une couche d'entrée (qui ne sert à rien, sinon à transmettre le signal d'entrée vers les couches supérieures) comportant d neurones. La fonction d'activation des "neurones" de cette couche est la plupart du temps une fonction linéaire de pente un.

- une couche de sortie comportant K neurones,

- un certain nombre de couches intermédiaires, dites couches cachées.

Une réseau de neurones peut donc être considéré comme une fonction de $\mathbb{R}^d$ dans $\mathbb{R}^K$, qui à un vecteur $\mathbf{x}$ associe un vecteur $(\mathbf{x}) = (d_k(\mathbf{x}))$. Dans ces notes nous nous limiterons au type de réseau le plus courant, comportant trois couches (c'est à dire une seule couche cachée) où chaque sortie y_k est de la forme

$$y_k = d_k\left(\alpha_k + \sum_{j\to k} w_{jk} \cdot d_j(\alpha_j + \sum_{i\to j} w_{ij} \cdot x_i)\right),$$

où $i \to j$ dénote l'ensemble des neurones i qui sont connecté au neurone j. Au même titre que dans la section précédente, il est avantageux d'un point de vue notation, d'augmenter la dimension du problème en ajoutant une composante unité à chaque entrée de neurone (voir figure 4.3). Cette astuce permet de ce débarrasser des terme de biais α_i et la sortie y_k prend alors la forme (figure 4.3) :

$$y_k = d_k\left(\sum_{j\to k} w_{jk} \cdot d_j(\sum_{i\to j} w_{ij} \cdot x_i)\right).$$

Ce genre de réseau est suffisamment général pour pouvoir approximer n'importe quelle fonction continue de $\mathbb{R}^d$ dans $\mathbb{R}^K$, de façon aussi précise que l'on veut.

4.2.3 Discrimination et réseaux à couches

Un réseau de neurones est une fonction () définie par de nombreux paramètres $\mathbf{w} = \{w_{ij}\}$, qui sont les poids de connexions. Dans le cadre de la discrimination, l'objectif consiste à ajuster les paramètres $\mathbf{w}$ de manière à transformer les K sorties du réseau en fonctions discriminantes $d_k(\mathbf{x})$, qui définiront un classifieur. Rappelons qu'un classifieur affecte le vecteur forme $\mathbf{x}$ à la classe k si :

$$d_k(\mathbf{x}) > d_\ell(\mathbf{x}),\ \forall \ell \neq k.$$

Il est possible de construire une fonction $\mathbf{z} \in \mathbb{R}^K$ qui définisse un classifieur idéal pour l'ensemble d'apprentissage $\mathcal{F} = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}_N))\}$. Considérons par exemple $\mathbf{z}(\mathbf{x}_i) = (z_k(\mathbf{x}_i), k = 1, ..., K)$ avec $z_k(\mathbf{x}_i) \in \{0, 1\}$, $\sum_{k=1}^K z_k(\mathbf{x}_i) = 1$, et $z_k(\mathbf{x}_i) = 1$ signifiant que $\mathbf{x}_i$ appartient à la classe k. L'idée originale de **?**) consiste

Figure 4.3: Réseau de neurone avec une seule couche cachée

à choisir le vecteur $\mathbf{w}$, de manière à minimiser le carré de l'erreur entre () et $\mathbf{z}$ sur l'ensemble d'apprentissage :

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(\mathbf{z}(\mathbf{x}_i) - (\mathbf{x}_i; \mathbf{w})\right)^2$$

La minimisation de ce critère rend aussi proche que possible, au sens des moindres carrés, le réseau de neurones () de la fonction cible $\mathbf{z}$.

Exemple 4.1 Soit un problème de discrimination à deux classes. Si $\mathbf{x}_i$ un vecteur forme de l'ensemble d'apprentissage appartient à la classe 2, alors la fonction cible $\mathbf{z}()$ vaudra en $\mathbf{x}_i$

$$\mathbf{z}(\mathbf{x}_i) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

et la sortie du réseaux sera de la forme

$$(\mathbf{x}_i; \mathbf{w}) = \begin{bmatrix} d_1(\mathbf{x}_i) \\ d_2(\mathbf{x}_i) \end{bmatrix}$$

△

La littérature connexioniste propose de nombreux critères d'erreur qui possèdent chacun certains avantages. Citons par exemple celui de **?**) :

$$E(\mathbf{w}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{\left(\mathbf{z}_k(\mathbf{x}_i) - d_k(\mathbf{x}_i; \mathbf{w})\right)^2}{1 - d_k(\mathbf{x}_i; \mathbf{w})^2}.$$

Les auteurs arguent que la minimisation de l'erreur quadratique, par un algorithme d'optimisation classique peut aboutir à des minima locaux dûs à la saturation des fonctions d'activation des neurones de sortie. En effet si, les poids des neurones de sortie sont très grands, alors leur fonction d'activation vaut un, et la fonction d'erreur est excessivement plate. Le critère précédent a pour but d'éviter ces zones dangereuses, en faisant croître l'erreur de manière très importante à l'approche de la saturation.

Notons qu'une autre solution très utilisée et efficace pour lutter contre la saturation (de toutes les unités cette fois-ci) consiste à optimiser un critère pénalisé :

$$E_p(\mathbf{w}) = E(\mathbf{w}) + \lambda \sum w_{ij}^2.$$

Avant d'utiliser ce type de critère, il faut s'assurer que les entrées et les sorties de chaque neurone sont à la même échelle. Ainsi une normalisation des vecteurs forme constitue dans ce cas une étape indispensable.

Citons enfin, l'entropie croisée qui est un critère très utilisé dans les applications de type discrimination :

$$E(\mathbf{w}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \left[z_k(\mathbf{x}_i) \log \frac{z_k(\mathbf{x}_i)}{d_k(\mathbf{x}_i; \mathbf{w})} + (1 - z_k(\mathbf{x}_i)) \log \frac{1 - z_k(\mathbf{x}_i)}{1 - d_k(\mathbf{x}_i; \mathbf{w})} \right]$$

Remarquons que cet dernier critère suppose que la fonction cible et les sortie du réseaux sont comprises entre 0 et 1.

4.2.4 Apprentissage

Une fois un critère d'erreur défini reste à l'optimiser. L'idée première de **?**) consistait à utiliser une descente de gradient :

$$w_{ij}^{q+1} = w_{ij}^{q} - \nu \cdot \frac{\partial E}{\partial w_{ij}}$$

L'application de cette technique nécessite donc le calcul explicite des dérivées partielles du critère d'erreur $E(\mathbf{w})$ par rapport à chacun des paramètres w_{ij} du réseau.

Rappelons que la fonction d'activation de chaque neurone j est une fonction dérivable de la somme pondérées x_j des entrées y_i :

$$y_j = d_j\left(\sum_{i \to j} w_{ij} y_i\right) = d_j(x_j).$$

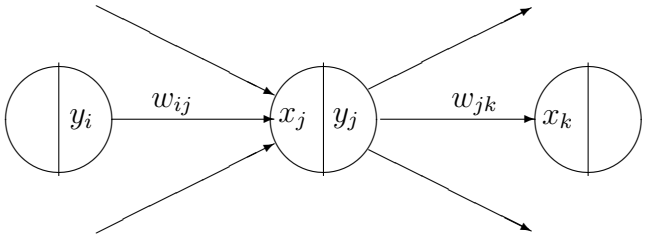

Figure 4.4: Notations utiles au calcul du gradient

Un rappel technique de la règle de différentiation des fonctions composées n'est peut être pas inutile pour comprendre la suite de cette section. Considérons les trois fonctions suivantes supposées continues et partiellement dérivables par rapport à toutes leurs variables :

$$\begin{aligned} u &= g_1(x, y) \\ v &= g_2(x, y) \\ h &= f(u, v) \end{aligned}$$

Les dérivées partielles de h par rapport à x et y s'expriment alors

$$\frac{\partial h}{\partial x} = \frac{\partial h}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial h}{\partial v}\frac{\partial v}{\partial x}, \tag{4.1}$$

$$\tag{4.2}$$

$$\frac{\partial h}{\partial y} = \frac{\partial h}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial h}{\partial v}\frac{\partial v}{\partial y}. \tag{4.3}$$

L'idée à la base du calcul du gradient de la fonction d'erreur E par rapport aux poids w_{ij}, consiste à exprimer la dérivée de E par rapport à des variables de plus en plus proche de la couche de sortie. Notons que le critère E, est une somme sur l'ensemble d'apprentissage des erreurs E_p commises pour chaque vecteur forme $\mathbf{x}_p$. On a donc

$$\frac{\partial E}{\partial w_{ij}} = \sum_p \frac{\partial E_p}{\partial w_{ij}}$$

En appliquant la règle de dérivation précédente au calcul qui nous intéresse, on obtient

$$\frac{\partial E_P}{\partial w_{ij}} = \frac{\partial E_P}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = y_i \cdot \frac{\partial E_P}{\partial x_j} = y_i \cdot \delta_j$$

En répétant l'opération sur δ_j, il vient

$$\delta_j = \frac{\partial E_P}{\partial x_j} = \frac{\partial E_P}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_j} = \frac{\partial E_P}{\partial y_j} \cdot d'_j(x_j).$$

Deux cas sont alors à distinguer :

- si le neurone j appartient à la couche de sortie alors la quantité $\frac{\partial E_P}{\partial y_j}$ est calculable. Considérons par exemple des neurones logistiques et une fonction d'erreur quadratique :
$$\delta_j = 2(y_j - z_j) \cdot yj \cdot (1 - y_j)$$
avec z_j la jcoordonnée de la fonction cible $\mathbf{z}(\mathbf{x}_p)$.

- si le neurone j fait partie de la couche cachée, alors il faut continuer à développer les δ_j de manière à les exprimer en fonction de variables relatives à la couche de sortie :
$$\frac{\partial E_P}{\partial y_j} = \sum_{k:\ j \to k} \frac{\partial x_k}{\partial y_j} \cdot \frac{\partial E_P}{\partial x_k} = \sum_{k:\ j \to k} w_{jk} \cdot \delta_k,$$
Pour comprendre cette égalité, remarquons qu'une modification de y_j va être répercutée sur E_P par l'intermédiaire de tous les neurones k auquel le neurone j transmet sa sortie y_j. Remarquons que dans le cas où le réseau considéré comporte plus d'une couche cachée, les calculs précédents doivent être réitérés afin de remonter aux quantités δ relatives à la couche de sortie.

L'algorithme de descente de gradient précédent est connu sous le nom de de rétropropagation du gradient. Chaque itération nécessite deux étapes :

- une passe avant qui détermine les sorties de chaque neurone en fonction de vecteurs forme présentés à la couche d'entrée ;

- une passe arrière, où les δ sont propagés de la couche de sortie vers la couche d'entrée, de manière à pouvoir calculer les différentes composantes du gradient.

De nombreuses autres techniques d'optimisation sont employées pour ajuster les poids des réseaux multicouches. Notons que les algorithmes neuronaux modifient souvent les paramètres du réseau en fonction de l'erreur commise pour un seul vecteur forme (algorithme "on-line").

Ces algorithmes d'optimisation posent plusieurs problèmes :

- où commencer ? En effet le résultat obtenu par l'algorithme d'optimisation dépend du point de départ, c'est-à-dire de l'initialisation des vecteurs poids du réseau.

- où s'arrêter ? Ce problème est lié à celui de la complexité du réseau. En effet, si un réseau est défini par de nombreux paramètres, l'agorithme d'optimisation utilisé est suceptible d'obtenir de très petites erreurs sur l'ensemble d'apprentissage. Ce phénomène n'est en général pas souhaitable, car il résulte de l'hyper-spécialisation du résau pour un ensemble d'apprentissage donné, ce qui entraine une mauvaise capacité à généraliser.

Le premier problème est en général résolu en initialisant les poids de connexions au hasard, tout en évitant les valeurs trop grandes qui posent des problème de saturation.

Le second problème peut être traité de trois manières différentes :

- une solution consiste simplement à stopper l'apprentissage avant que l'erreur en généralisation se dégrade ("early stopping") ; une technique répandue utilise un ensemble de validation. L'apprentissage, qui diminue l'erreur commise sur l'ensemble d'apprentissage, est alors stoppé lorsque l'erreur augmente sur cet ensemble de validation.

- il est aussi possible d'utiliser la régularisation, qui limite d'office la complexité du réseau ;

- ou enfin de recourir à un modèle plus simple, au quel cas l'on se ramène à un problème de choix de modèle.

4.2.5 Choix du nombre de neurones cachés

Dans le cadre que nous nous sommes fixés (une seule couche cachée) le nombre de neurones de la couche d'entrée est identique à la dimension des vecteurs formes, et le nombre de neurones de la couche de sortie correspond au nombre de classes. Par contre les connexions entre les différents neurones, ainsi que le nombre de neurones cachés restent des paramètres à choisir.

Ce dernier problème constitue typiquement un problème de choix de modèle. Les techniques usuelles peuvent être envisagées : comparaison de modèles deux à deux en utilisant une validation croisée, ou bien des critères de complexités comme AIC. Une démarche répandue consiste à comparer deux réseaux totalement connectés qui diffèrent par le nombre de neurones de la couche cachée. Les constructions

incrémentales, qui partent d'un petit réseau et ajoutent une nouvelle unité à chaque étape semblent être les plus efficaces, ou du moins les plus populaires.

Chapter 5

Extraction et sélection de caractéristiques

En reconnaissance des formes, les algorithmes de classement classent des vecteurs forme $\mathbf{x}_i$ qui sont des descriptions d'objets. Cette approche fait l'hypothèse qu'une description cohérente de l'ensemble des objets existe. Dans ce contexte, la question suivante émerge : comment choisir les caractéristiques descriptives. Ce chapitre envisage quelques réponses à cette interrogation.

Dans un premier temps, précisons le problème. Si l'on suppose que chaque objet est caractérisé par d variables quantitatives (poids, taille, vitesse...), et appartient à une classe parmi K, le problème précédent peut alors se reformuler comme suit :

- quelles variables parmi les d disponibles sont les plus discriminantes ?
- comment extraire de nouvelles variables, à partir des variables initiales, qui soient discriminantes ?

Utilisées comme prétraitement des données, ces deux approches, extraction et sélection de caractéristiques, permettent de faciliter la tâche du classifieur. La réduction de dimension (passage de d à q variables avec $q < d$) est par exemple d'une grande aide pour les méthodes non paramétriques de classement basées sur les estimateurs à noyaux (chapitre 3). En effet, ces dernières méthodes sont inefficaces dans les espaces de grandes dimensions et gagnent à être utilisées après réduction de dimension.

Dans le cadre d'une analyse exploratoire des données, les techniques d'extraction et de sélection de caractéristiques permettent aussi de visualiser les vecteurs forme.

Dans ce chapitre nous présentons :

- l'analyse factorielle discriminante, qui est une technique d'extraction de caractéristiques cherchant des combinaisons linéaires des variables initiales ;

- quelques méthodes de positionnement multidimensionel ("multidimensional scaling"), qui ne sont pas spécifique au problème du classement, mais permettent la visualisation des vecteurs forme ;
- les principales méthodes de sélection de caractéristiques.

5.1 Analyse factorielle discriminante

L'analyse factorielle discriminante vise à trouver des nouvelles variables, appelées facteurs, combinaisons linéaires des variables initiales, qui permettent de distinguer le mieux possible, les K groupes de vecteurs forme. Historiquement la méthode a été introduite par **?**) dans le cas de deux classes. La généralisation au cas multi-classes est due à **?**)

Précisons quelques notations utiles à la présentation de la méthode :

- les vecteurs moyennes des K groupes sont notés
$$\mathbf{m}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{x}_{i,k}.$$
- $\mathbf{m}$ dénote le vecteur moyenne de l'ensemble des vecteurs formes,
$$\mathbf{m} = \frac{1}{N} \sum_{k=1}^{n_k} n_k \cdot \mathbf{m}_k,$$
- la matrice d'inertie intra-classe ${}_W$ (W comme "within") est définie par :
$${}_W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{i,k} - \mathbf{m}_k)(\mathbf{x}_{i,k} - \mathbf{m}_k)^t,$$
- la matrice d'inertie inter-classe ${}_B$ (B comme "between") est définie par :
$${}_B = \sum_{k=1}^{K} n_k \cdot (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^t,$$
- et enfin la matrice d'inertie totale ${}_T$ s'écrit :
$$\begin{aligned}
{}_T &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{i,k} - \mathbf{m})(\mathbf{x}_{i,k} - \mathbf{m})^t \\
&= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{i,k} - \mathbf{m}_k + \mathbf{m}_k - \mathbf{m})(\mathbf{x}_{i,k} - \mathbf{m}_k + \mathbf{m}_k - \mathbf{m})^t \\
&= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{i,k} - \mathbf{m}_k)(\mathbf{x}_{i,k} - \mathbf{m}_k)^t + \sum_{k=1}^{K} n_k \cdot (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^t \\
&= {}_W + {}_B .
\end{aligned}$$

  Notons que cette dernière égalité est une application du théorème de Huygens et une généralisation de la formule de l'analyse de la variance.

5.1.1 Discriminant linéaire de Fisher

Supposons un ensemble d'apprentissage $\mathcal{F}$ composé de n_1 vecteurs forme $\mathbf{x}_{i,1}$ de la classe $\mathcal{C}_1$ et n_2 vecteurs forme $\mathbf{x}_{j,2}$ de la classe $\mathcal{C}_2$. Une combinaison linéaire y d'un vecteur $\mathbf{x}$

$$y = \mathbf{w}^t \cdot \mathbf{x},$$

est un scalaire. Dans le but de poser un problème possédant une solution unique, on impose $\|\mathbf{w}\| = 1$.

Le but de l'analyse discriminante de Fisher est de trouver un facteur $\mathbf{w}$ tel que les sur laquelle les projections des vecteurs $\mathbf{x}_{i,1}$ soit bien séparées des projections des vecteurs $\mathbf{x}_{j,2}$.

Notons :

- $y_{i,k}$ les projections des $\mathbf{x}_{i,k}$: $y_{i,k} = \mathbf{w}^t \cdot \mathbf{x}_{i,k}$,
- $\tilde{m}_k$ les projections des vecteurs moyennes : $\tilde{m}_k = \mathbf{w}^t \cdot \mathbf{m}_k$,
- et $\tilde{s}_k^2$ l'inertie des projections des vecteurs formes du groupe k :

$$\tilde{s}_k^2 = \sum_{i=1}^{n_k} (y_{i,k} - \tilde{m}_k)^2.$$

Le critère mesurant la qualité de cette séparation sur l'axe $\mathbf{w}$, proposé par Fisher, est le rapport de l'inertie inter-classes sur l'inertie intra-classes des projections des vecteurs forme de $\mathcal{F}$:

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

Intuitivement les deux groupe projetés seront bien séparés, si l'écart entre $\tilde{m}_1$ et $\tilde{m}_2$ est grand, c'est-à-dire si les centre de gravité des groupes sont les plus éloignés possibles en projection, et si chaque classe projetée forme le groupe le plus compact possible autour de leur centre de gravité respectif (inertie intra-classe petite). En d'autres termes, la séparation sera d'autant plus nette que le critère $J(\mathbf{w})$ sera important.

Dans le contexte des tests d'hypothèses, lorsque l'on désire tester l'égalité des moyennes $\tilde{m}_1$ et $\tilde{m}_2$ de deux groupes, l'analyse de la variance amène à définir la région critique suivante :

$$J = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} > A.$$

On peut montrer que J est proportionnelle à une variable aléatoire qui suit une loi de Fisher Snedecor $F_{K-1,N-K}$. La maximisation du critère $J(\mathbf{w})$ revient donc à trouver un sous espace de projection où le test d'égalité des moyennes sera le plus pessimiste possible.

La recherche du vecteur $\mathbf{w}$ peut donc être formulé comme un problème d'optimisation sous contrainte :

$$\begin{cases} \mathbf{w} = arg\max J() \\ \|\mathbf{w}\| = 1 \end{cases}$$

En faisant apparaître dans le critère $J(\mathbf{w})$ les vecteurs forme $\mathbf{x}_{i,k}$, il vient

$$\begin{cases} \mathbf{w} = arg\max \frac{^t(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^t}{\frac{t}{W}} \\ \|\mathbf{w}\| = 1 \end{cases}$$

Remarquons que

$$_B = \frac{n_1 n_2}{N^2}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t. \tag{5.1}$$

En effet si l'on considère que les vecteurs forme sont centrés ($\mathbf{m} = 0$), on a

$$\mathbf{m} = \frac{n_1}{N}\mathbf{m}_1 + \frac{n_2}{N}\mathbf{m}_2 = 0.$$

On peut donc écrire que

$$\begin{cases} _B = \frac{n_1}{N}\mathbf{m}_1\mathbf{m}_1^t + \frac{n_2}{N}\mathbf{m}_2\mathbf{m}_2^t - \underbrace{\left(\frac{n_1}{N}\mathbf{m}_1\mathbf{m}_1^t + \frac{n_2}{N}\mathbf{m}_2\mathbf{m}_1^t\right)}_{0} = -\frac{n_2}{N}\mathbf{m}_2(\mathbf{m}_1 - \mathbf{m}_2)^t \\ \\ _B = \frac{n_1}{N}\mathbf{m}_1\mathbf{m}_1^t + \frac{n_2}{N}\mathbf{m}_2\mathbf{m}_2^t - \underbrace{\left(\frac{n_1}{N}\mathbf{m}_1\mathbf{m}_2^t + \frac{n_2}{N}\mathbf{m}_2\mathbf{m}_2^t\right)}_{0} = \frac{n_1}{N}\mathbf{m}_1(\mathbf{m}_1 - \mathbf{m}_2)^t \end{cases}$$

En moyennant (n_1 fois la première expression plus n_2 fois la deuxième) on montre l'équation **??** qui nous permet de poser le problème de la recherche de $\mathbf{w}$ sous la forme :

$$\begin{cases} \mathbf{w} = arg\max \frac{\frac{t}{B}}{\frac{t}{W}} \\ \|\mathbf{w}\| = 1 \end{cases}$$

On voit dans que λ_{max} la valeur maximum du critère vérifie :

$$\begin{aligned} \lambda_{max} \cdot \mathbf{w}_W^t\mathbf{w} &= \mathbf{w}_B^t\mathbf{w} \\ \lambda_{max} \cdot_W \mathbf{w} &= _B\mathbf{w} \end{aligned}$$

Si la matrice $_W$ est inversible, résoudre ce problème revient à chercher $\mathbf{w}$ le vecteur propre associé à λ_{max} la plus grande valeur propre de la matrice $\frac{-1}{W}{}_B$:

$$\lambda_{max} \cdot \mathbf{w} = \frac{-1}{W}{}_B \cdot \mathbf{w}.$$

Remarquons que la matrice $_B$ est de rang 1, ce qui implique qu'il n'existe qu'une seule valeur propre. Dans le cas qui nous intéresse on peut aisément obtenir le vecteur propre en remarquant que $S_B \cdot \mathbf{w}$ est toujours dans la direction du vecteur $\mathbf{m}_1 - \mathbf{m}_2$:

$$\mathbf{w} = \alpha_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2).$$

avec α un scalaire tel que $\|\mathbf{w}\| = 1$.

5.1.2 Cas général

La généralisation du discriminant linéaire de Fisher au cas multi-classes recherche un espace vectoriel de dimension p avec $p < K - 1$ telle que les K classes soient séparés au mieux dans cet espace. Le but de l'analyse factorielle discriminante consiste donc à trouver p vecteurs $\mathbf{w}_k$, qui définissent p variables discriminantes :

$$y_k = \mathbf{w}_k \cdot \mathbf{x},$$

L'ensemble des vecteurs $\mathbf{w}_k$, les facteurs, peut s'écrire de manière condensé sous la forme d'une matrice de dimension d par p (chaque colonne de cette matrice correspond à un des facteurs).

Notons :

- $$\tilde{\mathbf{m}}_k = \frac{1}{n_k} \sum_{i=1} n_k \mathbf{y}_{ik} = \frac{1}{n_k} \sum_{i=1} n_k \mathbf{y}_{ik} =^t \mathbf{m}_k,$$

  les vecteurs moyennes des nouvelles variables ;

- $$\tilde{\mathbf{m}} =^t \mathbf{m},$$

  le vecteur moyenne de l'ensemble des vecteurs forme, exprimé en fonction des facteurs ;

- $$\begin{aligned} \tilde{}_W &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{y}_{ik} - \tilde{\mathbf{m}}_k)(\mathbf{y}_{ik} - \tilde{\mathbf{m}}_k)^t \\ &= {}^t_W, \end{aligned}$$

  la matrice de d'inertie intra-classes des nouvelles variables ;

- $$\begin{aligned} \tilde{}_B &= \sum_{k=1}^{K} n_k \cdot (\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})^t, \\ &= {}^t_B, \end{aligned}$$

  la matrice d'inertie inter-classes des nouvelles variables.

Pour obtenir les vecteur $\mathbf{w}_k$ composant la matrice , il est possible de procéder par construction. Le vecteur $\mathbf{w}_1$, appelé premier facteur discriminant, est la solution du problème :

$$\begin{cases} \mathbf{w}_1 = arg \max \frac{{}^t_B}{{}^t_W} \\ \|\mathbf{w}_1\| = 1 \end{cases}$$

Dans la section précédente nous avons vu que la solution de ce problème était le vecteur propre de $\bar{}^{1}_{W}{}_{B}$ associé à la plus grande valeur propre. Le second facteur, s'il existe, est défini comme le vecteur propre associé à la deuxième plus grande valeur propre.

En poursuivant cette construction, on trouve autant d'axes discriminants que de valeurs propres non nulles, c'est à dire au plus $K-1$, le rang maximum de la matrice $\bar{}^{1}_{W}{}_{B}$. Remarquons que les vecteurs propres ainsi trouvés sont $_{W}$ orthogonaux :

$$\mathbf{w}_{\ell}^{t} \cdot_{W} \cdot \mathbf{w}_{k} = 0$$

si $\mathbf{w}_k$ et $\mathbf{w}_\ell$ sont deux facteurs distincts.

Il semble aussi intéressant de savoir quel critère optimise ce procédé de construction. À chaque étape, on cherche un vecteur $\mathbf{w}_k$ différent de tous les vecteurs trouvés précédemment, qui maximise le critère

$$\begin{aligned} J_k &= \frac{\mathbf{w}_{kB}^{t}\mathbf{w}_k}{\mathbf{w}_{kW}^{t}\mathbf{w}_k} \\ &= \mathbf{w}_{kW}^{t}{}^{-1}{}_{B}\mathbf{w}_k \\ &= trace\left[\mathbf{w}_k\mathbf{w}_{kW}^{t}{}^{-1}{}_{B}\right] \end{aligned}$$

Ainsi à la fin de la construction, on peut affirmer que la somme des critères J_k est maximisée (car chacun des termes de la somme est maximisée) sous la contrainte que les vecteurs $\mathbf{w}_k$ sont orthogonaux entre eux. La matrice est donc solution de :

$$\begin{cases} \max \sum_{k=1}^{p} trace\left[\mathbf{w}_k\mathbf{w}_{kW}^{t}{}^{-1}{}_{B}\right] = \max trace\left[\bar{}^{t}_{W}{}^{-1}_{B}\right] \\ {}^{t}\cdot_{W}\cdot = I \end{cases}$$

Ce critère met en évidence les liens existant avec l'analyse en composantes principale.

Remarquons qu'il est possible de montrer que la matrice est aussi solution de

$$\begin{cases} \max \frac{det(\tilde{}_{B})}{det(\tilde{}_{W})} = \frac{det(\tilde{}^{t}_{B})}{det(\tilde{}^{t}_{W})} \\ \tilde{}^{t}_{W} = I \end{cases}$$

Ce dernier critère peut s'interprèter comme le rapport entre deux volumes. Ce rapport sera important lorsque les classes projetées occuperons en moyenne un petit volume autour de leur moyenne ($det(\tilde{}_{W})$ petit), et que les vecteurs moyennes projetés occuperons occuperons un grand volume par rapport au précédent. Ceci correspond intuitivement bien à la notion de séparation que l'on recherche.

Cette formulation du problème en terme de critère globale a le défaut d'admettre plusieurs solutions dont la solution obtenue par construction. En effet toutes les transformation linéaires non singulière de laisse les critères invariants.

Notons que la matrice peut être trouver de deux autre manières. Il est en effet direct de montrer que :

$$arg\max_{\mathbf{w}} \frac{\mathbf{w}_{B}^{t}\mathbf{w}}{\mathbf{w}_{W}^{t}\mathbf{w}} = arg\max_{\mathbf{w}} \frac{\mathbf{w}_{T}^{t}\mathbf{w}}{\mathbf{w}_{W}^{t}\mathbf{w}} = arg\max_{\mathbf{w}} \frac{\mathbf{w}_{B}^{t}\mathbf{w}}{\mathbf{w}_{T}^{t}\mathbf{w}}.$$

En utilisant le processus de construction précédent à partir de ces critères, on trouve la même matrice . Il s'ensuit que les matrices $(\overline{}_W^{-1}\cdot_B)$, $(\overline{}_W^{-1}\cdot_T)$ et $(\overline{}_T^{-1}\cdot_B)$ ont même vecteurs propres.

5.2 Multidimensional scaling

Pour réduire la dimension des vecteurs forme, une alternative consiste à utiliser des techniques de multidimensionnal scaling. Ces techniques sont très utilisées en psychologie et sociologie, où les objets d'étude (les individus) ne sont pas décrit en terme de caractéristiques individuelles mais les uns par rapport aux autres par la mesure de leur différence deux à deux. Ces mesures sont appelées dissimilarités et non distances car elles ne vérifient pas l'inégalité triangulaire.

Les méthodes de multidimensional scaling visent à représenter au mieux des objets dans un espace visualisable, de façon à ce que les distances entre ces objets dans cet espace soient aussi proches que possible des dissimilarités initiales.

Les dissimilarités mesurent les différences entre toutes les paires d'objets. Ainsi les dissimilarités entre N objets sont spécifiées par une matrice $N \times N$, $\delta = \{\delta_{ij}\}_{i,j=1..N}$. Ce genre de matrices peut avoir différentes origines :

- Le jugement humain peut souvent être traduit par des mesures de dissimilarités. Une personne amenée à quantifier la différence de confort entre deux voitures pourra, par exemple, choisir une chiffre entre 1 et 10 qui correspond à l'intensité de la différence perçue.

- Les données telles que les temps de transports entre des paires de villes se présentent naturellement sous la forme d'une matrice de dissimilarités.

- Enfin, notons qu'il est toujours possible de dériver une matrice de dissimilarités (les distances sont des dissimilarités) d'un ensemble de vecteurs forme. Il suffit en effet de calculer les distance entre tous les vecteurs forme. C'est cette dernière approche qui justifie la présence de cette section dans ces notes.

Parmi les techniques de positionnement multidimensionel, les approches métriques sont couramment opposées aux approches non métriques. Les premières produisent des représentations préservant au mieux l'information quantitative (**?**) contenue dans les données, alors que les secondes privilégient l'information qualitative (**?**).

5.2.1 Projection de Sammon

La projection de Sammon est une méthode métrique très populaire. Cette méthode cherche à "projeter" non linéairement les vecteurs formes $\mathbf{x}_1, \cdots, \mathbf{x}_N$ dans un espace de plus faible dimension ($\mathbb{R}^1$, $\mathbb{R}^2$, ou bien $\mathbb{R}^3$). Notons

- $\mathbf{y}_1, \cdots, \mathbf{y}_N$ les représentations des $\mathbf{x}_i$ dans cet espace de faible dimension,

- δ_{ij}, la distance (ou bien dissimilarité) entre les vecteurs formes $\mathbf{x}_i$ et $\mathbf{x}_j$,
- d_{ij}, la distance (euclidienne ou autre) entre $\mathbf{y}_i$ et $\mathbf{y}_j$.

Le critère proposé par **?**) pour juger de la qualité de la représentation est le suivant :

$$S = \frac{1}{\sum_{i>j} \delta_{ij}} \sum_{i>j} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}, \tag{5.2}$$

Remarquons que ce critère prend en compte les erreurs commises sur les petites distances, contrairement à une ACP. Il est normalisé de manière à être invariant pour des rotations, translations et changements d'échelles.

La recherche de la configuration optimale des $\mathbf{y}_i$ peut s'effectuer par une descente de gradient

$$\mathbf{y}_k^{q+1} = \mathbf{y}_k^q - \alpha_q \nabla_{y_k}^t S$$

Par exemple, si d_{ij} est la distance euclidienne, le gradient du critère par rapport au point $\mathbf{y}_k$ prend la forme suivante :

$$\nabla_{y_k}^t S = \frac{1}{\sum_{i>j} \delta_{ij}} \sum_{j \neq k} \frac{(d_{kj} - \delta_{kj})}{\delta_{kj}} \cdot \frac{(\mathbf{y}_k - \mathbf{y}_j)}{d_{kj}}$$

La configuration initiale des $\mathbf{y}_i$ peut être choisie au hasard mais le processus semble converger plus vite si l'on part d'une solution approchée (celle obtenue par ACP par exemple.)

5.2.2 Projection de Kruskal

L'approche non métrique favorise les représentations qui privilégient l'ordre relatif entre les dissimilarités initiales plutôt que les valeurs. Reprenons les notations utilisées pour la projection de Sammon, en y ajoutant les $\hat{d}_{ij}$ qui sont des nombres vérifiant la contrainte suivante : si les dissimilarités initiales sont ordonnées

$$\delta_{i_1 j_1} \leq \cdots \leq \delta_{i_N j_N},$$

alors on a

$$\hat{d}_{i_1 j_1} \leq \cdots \leq \hat{d}_{i_N j_N}.$$

?) propose de trouver une configuration des $\mathbf{y}_i$ telle que les distances $d_{ij} = dist(\mathbf{y}_i, \mathbf{y}_j)$ soient optimales au sens du critère

$$S = \sqrt{\frac{S^*}{T^*}} = \sqrt{\frac{\sum_{i>j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i>j} d_{ij}^2}}$$

L'optimisation de ce critère est plus délicat que celui de Sammon. Remarquons d'ailleurs que bien que l'article de Sammon soit postérieur de cinq ans, son critère et

la méthode d'optimisation proposée constituent un cas particulier de l'approche de Kruskal.

Kruskal propose une méthode d'optimisation alternée pour optimiser les critères. Chaque itération se partage en deux étapes :

- une régression isotonique est utilisée pour calculer les $\hat{d_{ij}}$ qui minimisent le critère (on considère les d_{ij} fixés), et respectent la contrainte de montonicité.

- le gradient du critère par rapport à chaque $\mathbf{y}_i$ est calculé, et une nouvelle configuration $\mathbf{y}_1, \cdot, \mathbf{y}_N$ est obtenue en modifiant les points dans la direction du gradient (un pas d'une descente de gradient).

5.3 Sélection de caractéristiques

Dans cette section sont présentées quelques méthodes qui permettent de sélectionner q variables parmi les d disponibles. Le but consiste à garder les variables les plus discriminantes, c'est-à-dire, les variables qui pour un classifieur donné vont produire le taux d'erreur le plus petit possible. Ce problème peut être partagé en deux sous problèmes distincts :

- d'une part, il faut disposer d'un critère pour juger la qualité d'un groupe de variables,

- d'autre part, il est nécessaire d'utiliser des heuristiques pour optimiser ce critère. En pratique, il est en effet souvent impossible d'explorer tous les groupe des q variables parmi d.

5.3.1 Critères de choix

Le critère, qui mesure la performance d'un classifieur, est le risque total E^*, qui est équivalent à la probabilité d'erreur dans le cas où le coût $\{0, 1\}$ est utilisé. Ce critère paraît constituer un choix logique pour juger de la qualité d'un groupe de variables, mais en pratique, de nombreuses autres mesures plus rapides à estimer ont été proposées :

- $C_1 = trace\left[{}^{-1}_{W_q B_q}\right] = \sum_{i=1}^{q} \lambda_i$. Ce critère vient de l'analyse factorielle discriminante. Plus il est grand et plus le groupe de q variables considéré est discriminant au sens de l'AFD. C'est donc une mesure de la séparabilité linéaire para rapport au groupe de variables.

- $C_2 = \frac{|W_q|}{|T_q|} = \prod_{i=1}^{q} \frac{1}{1+\lambda_i}$ (lambda de Wilks). Comme précédemment ce critère peut s'interpréter dans le cadre de l'AFD. Remarquons que minimiser C_2 revient à maximiser C_1.

- $C_3 = [(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)]^{\frac{1}{2}}$ (distance de Mahalanobis). Dans le cadre limité de la discrimination linéaire entre deux classes, l'erreur commise ne dépend que de cette distance (Chapitre 2, exemple 1.4). Notons que dans ce cas de figure, C_1, C_2 et C_3 sont équivalents.

- $C_4 = \sqrt{p_1 p_2} \int \sqrt{f_1(\mathbf{x}) f_2(\mathbf{x})} d\mathbf{x}$ (distance de Bhattacharya). Dans le cas de deux classes, le risque de Bayes est borné supérieurement par cette distance :

$$\begin{aligned} E^* &= I\!E\left[R(\hat{c}(X)|X)\right] \\ &= \int \min\left(\pi(1|\mathbf{x}), \pi(2|\mathbf{x})\right) f(\mathbf{x}) d\mathbf{x} \\ &= \int \min\left(p_1 f_1(\mathbf{x}), p_2 f_2(\mathbf{x})\right) d\mathbf{x} \\ &\leq \sqrt{p_1 p_2} \int \sqrt{f_1(\mathbf{x}) f_2(\mathbf{x})} d\mathbf{x} \end{aligned}$$

  car $min(a, b) \leq a^s b^{1-s}$. Remarquons encore que dans le cas de la discrimination linéaire, ce critère est équivalent aux trois précédents.

5.3.2 Procédures de sélection

Les heuristiques de sélection de variables comparent à chaque itération deux groupes distincts par rapport à la valeur d'un critère, et retiennent le meilleur groupe au sens de ce critère.

La sélection ascendante commence par considérer toutes les variables séparement. La meilleure variable (au sens du critère choisi) est sélectionnée lors de la première étape. La seconde étape cherche quelle variable ajoutée à la première produit la meilleure valeur du critère. Ce processus de construction est poursuivi jusqu'à obtenir les q variables souhaitées. Notons que cette heuristique est optimale à chaque étape, mais ne garantit pas que le groupe de q variables final soit le meilleur. Ceci est vrai seulement si le critère peut se décomposer en somme ou produit dont chaque terme ou facteur est fonction d'une seule variable.

La sélection descendante commence par considérer toutes les variables, et les élimine une par une. La variable éliminée à chaque étape est évidemment la moins bonne au sens du critère.

Chapter 6

Classification automatique

Une confusion répandue existe entre les termes classement et classification (respectivement "classification" et "clustering" en anglais). Le classement présuppose l'existence de classes dont certains objets sont connus, alors que la classification tente de découvrir une structure de classes qui soit "naturelle" aux données. Dans la littérature liée à la reconnaissance des formes, la distinction entre les deux approches est souvent désignée par les termes "apprentissage supervisé" et "non supervisé". Une classification peut avoir différentes motivations : compresser des informations, décrire de manière simplifiée de grandes masses de données, structurer un ensemble de connaissances, révéler des structures, des causes cachées, réaliser un diagnostic...

Dans le contexte du classement, les fonctions discriminantes définissent la similitude entre un vecteur forme et les différentes classes (chapitre 4). Ces fonctions discriminantes peuvent être construite à partir d'une mesure de ressemblance entre les vecteurs forme. En classification le même type d'approche est à la base des méthodes, et l'étape préalable à toute classification consiste à définir une mesure de ressemblance entre les objets (vecteurs formes). Traditionnellement deux démarches sont envisagées :

- on peut dire que deux objets sont semblables s'ils partagent une certaine caractéristique. Considérons le nombre de doigts d'un être vivant et comparons le singe et l'homme : sur ce critère de comparaison (et sur bien d'autres) les deux espèces seront jugées semblables. Ce genre de démarche aboutit à une classification *monothétique* base de l'approche aristotélicienne (**?**). Tous les objets d'une même classe partagent alors un certain nombre de caractéristiques (e.g. : "Tous les hommes sont mortels") ;

- on peut aussi mesurer la ressemblance en utilisant une mesure de proximité (distance, dissimilarité). Dans ce cas la notion de ressemblance est mesurée de façon plus floue et deux objets d'une même classe posséderont des caractéristiques

"proches" au sens de la mesure utilisée. Cette démarche est dite *polythétique*.

Le contexte de ce chapitre est l'approche polythétique et plus particulièrement, les méthodes de classification qui mesurent la ressemblance à l'aide d'une distance..

Une classification amène à répartir l'ensemble des vecteurs forme en différentes classes *homogènes*. La définition d'une classe et les relations entre classes peuvent être très variées. Dans ce chapitre nous nous intéresserons aux deux principales structures de classification :

- la partition,
- la hiérarchie.

6.1 Partitions

Définition 6.1 *$\mathcal{F}$ étant un ensemble fini, un ensemble $P = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K)$ de parties non vides de $\mathcal{F}$ est une partition si :*

1. $\forall i \neq j$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$,
2. $\cup_i \mathcal{C}_i = \mathcal{F}$.

Dans un ensemble $\mathcal{F} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ partitionné en K classes, chaque élément de l'ensemble appartient à une classe et une seule. Une manière pratique de décrire cette partition P consiste à utiliser une notation matricielle. Soit $\mathbf{c}(P)$ la matrice caractéristique de la partition $P = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K)$ (ou matrice de classification) :

$$\mathbf{c}(P) = \mathbf{c} = \begin{pmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NK} \end{pmatrix}$$

où $c_{ik} = 1$ si et seulement si $\mathbf{x}_i \in \mathcal{C}_k$, et $c_{ik} = 0$ sinon. Remarquons que la somme de la $i$ligne est égale à 1 (un élément appartient à une seule classe) et la somme des valeurs de la $k$colonne vaut n_k le nombre d'éléments de la classe $\mathcal{C}_k$. On a donc $\sum_{k=1}^{K} n_k = N$.

La notion de partition dure repose sur une conception ensembliste classique. Considérant les travaux de **?**) sur les ensembles flous, une définition du concept de partition floue semble "naturelle". La classification floue, développée au début des années 1970 (**?**), généralise une approche classique en classification en élargissant la notion d'appartenance à une classe.

Dans le cadre de la conception ensembliste classique, un individu $\mathbf{x}_i$ appartient appartient ou n'appartient pas à un ensemble donné $\mathcal{C}_{\|}$. Dans la théorie des sous-ensembles flous, un individu peut appartenir à plusieurs classes avec différent degrés d'appartenance. En classification cela revient autoriser les vecteurs formes à appartenir à toutes les classes, ce qui se traduit par le relâchement de la contrainte de

binarité sur les coefficients d'appartenance c_{ik}. Une partition floue est définie par une matrice de classification floue $\mathbf{c} = \{c_{ik}\}$ vérifiant les conditions suivantes :

1. $\forall k = 1..K$, $\forall \mathbf{x}_i \in \mathcal{F}$, $c_{ik} \in [0, 1]$.
2. $\forall k = 1..K$, $0 < \sum_{i=1}^{N} c_{ik} < N$,
3. $\forall \mathbf{x}_i \in \mathcal{F}$, $\sum_{k=1}^{K} c_{ik}$.

La seconde condition traduit le fait qu'aucune classe ne doit être vide et la troisième exprime le concept d'appartenance totale.

6.1.1 Critères et algorithmes

Les concepts de partition et de classification polythétique étant précisés, la question suivante émerge : comment trouver une partition optimale d'un ensemble de données, lorsque la ressemblance entre deux individus est évaluée par une mesure de proximité ?

La première chose à faire consiste à clarifier formellement le sens du mot optimal. La solution généralement adoptée est de choisir une mesure numérique de la qualité d'une partition. Cette mesure est parfois appelée critère, fonctionnelle, ou bien encore fonction d'énergie. L'objectif d'une procédure de classification est donc de trouver la partition ou les partitions qui donnent la meilleure valeur (la plus petite ou la plus grande) pour un critère donné.

Mais le nombre de partitions possibles, même pour un problème de taille raisonnable, est énorme. En effet si l'on considère un ensemble de N objets à partitionner en K classes, le nombre de partitions possibles est :

$$NP(N, K) = \frac{1}{K!} \sum_{k=0}^{K} (-1)^{k-1} \cdot C_k^K \cdot k^N. \tag{6.1}$$

Exemple 6.1 Soit un ensemble de 8 objets que l'on désire partager en 4 classes. Il existe 1701 partitions possibles !

△

Plutôt que de chercher la meilleure partition, celle qui donne la valeur optimale du critère, on utilise des méthodes plus rapides qui convergent vers des optima "locaux" du critère. Les partitions ainsi trouvées sont souvent satisfaisantes.

6.1.2 Inertie intra-classe et partition

De nombreux critères existent (**?**). Certains peuvent être liés, comme nous le verrons dans la suite, au choix d'un modèle pour l'ensemble des données. L'une des fonctions les plus utilisée est la somme des variances intra-classes :

$$\begin{aligned} I_W &= \sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik}\|\mathbf{x}_i - \mathbf{m}_k\|^2 \\ &= trace(_W) \end{aligned}$$

où les $\mathbf{m}_k$ sont les prototypes (centres) des classes et les c_{ik} sont les éléments d'une matrice de partition dure. Le problème posé est alors un problème d'optimisation sous contraintes (liées aux c_{ik}) :

$$(\hat{\mathbf{c}}, \hat{\mathbf{m}}) = arg \min_{(\mathbf{c},\mathbf{m})} I_W(\mathbf{c}, \mathbf{m}) \tag{6.2}$$

où $\mathbf{m}$ représente l'ensemble des centres de gravités.

L'algorithme des centres mobiles

Un algorithme très répandu pour résoudre ce problème est celui des *k-means* ou centres mobiles. Historiquement, cet algorithme date des années soixante. Il a été proposé par plusieurs chercheurs dans différents domaines à des dates proches (**?**, **?**). Cet algorithme basé sur des considérations géométriques doit certainement son succès à sa simplicité et son efficacité :

1. Initialisation des centres : une méthode répandue consiste à initialiser les centres avec les coordonnées de K points choisis au hasard.

2. Ensuite les itérations possèdent la forme alternée suivante :

   (a) étant donné $\mathbf{m}_1, \cdots, \mathbf{m}_K$, choisir les c_{ik} qui minimisent I_W,

   (b) étant donné $\mathbf{c} = \{c_{ik}\}$, minimiser I_W par rapport aux prototypes $\mathbf{m}_1, \cdots, \mathbf{m}_K$.

La première étape affecte chaque $\mathbf{x}_i$ au prototype le plus proche, et la seconde étape recalcule la position des prototypes en considérant que le prototype de la classe i devient son vecteur moyenne. Il est possible de montrer que chaque itération fait décroître le critère mais aucune garantie de convergence vers un maximum global n'existe en général. Si le critère des k-means est considéré du point de vue de la recherche d'une partition floue, c'est-à-dire si les contraintes sur les c_{ik} sont relâchées et deviennent $c_{ik} \in [0, 1]$ à la place de $c_{ik} \in \{0, 1\}$, la partition optimale au sens du nouveau critère est celle qui est optimale pour le critère classique (**?**). En d'autre terme, il n'y a aucun intérêt à considérer des partitions floues, lorsqu'on travaille avec le critère des k-means.

Cette forme d'algorithme alterné où un certain critère est optimisé, alternativement par rapport aux variables d'appartenance aux classes, puis par rapport aux paramètres définissant ces classes a été intensivement exploité. Citons entre autre *les nuées dynamiques* de Diday (**?**) et l'algorithme des *fuzzy c-means* (**?**).

Notons que Webster **?**) (à ne pas confondre avec Ronald Fisher) avait proposé un algorithme trouvant la partition optimale, au sens de la variance intra-classe, d'un ensemble de N données unidimensionnelles en $O(N \cdot K^2)$ opérations en utilisant des méthodes issues de la programmation dynamique.

Une version adaptative des centres mobiles

Une autre version des *k-means* (**?**) consiste à modifier les prototypes des classes en considérant les données une à une. On parle alors d'algorithme adaptatif :

1. Les K prototypes sont tirés au hasard parmi les N points.
2. A l'itération q, un individu $\mathbf{x}_i$ est choisi au hasard.
   - Détermination du prototype le plus proche de $\mathbf{x}_i$:
     $$\mathbf{m}_k^q = \min_j \|\mathbf{x}_i - \mathbf{m}_j^q\|.$$
     L'individu est affecté à la classe k.
   - Modification du prototype $\mathbf{m}_k^q$:
     $$\mathbf{m}_k^{q+1} = \frac{\mathbf{x}_i + n_k^q \cdot \mathbf{m}_j^q}{n_k^q + 1},$$
     et
     $$n_k^{q+1} = n_k^q + 1$$
     où n_k^q représente l'effectif de la classe k à l'itération q.

Les algorithmes adaptatifs sont particulièrement adéquats lorsque toutes les données à classer ne sont pas disponibles à l'avance. Les paramètres définissant les classes peuvent alors être ajustés à l'apparition de chaque nouvelle donnée sans trop de calculs.

Les nuées dynamiques

L'algorithme des nuées dynamiques (**?**) est une généralisation de centres mobiles. L'idée de base consiste à remplacer les prototypes $\mathbf{m}_k$ (vecteurs de $\mathbb{R}^d$) par des éléments de nature très diverse, nommés noyaux. Le noyau $\mathbf{n}_k$ d'une classe peut être par exemple, un ensemble de vecteurs forme de l'ensemble d'apprentissage, une droite, une loi de probabilité... La partition est obtenue en minimisant un critère de la forme

$$W(\mathbf{n}, \mathbf{c}) = \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} D(\mathbf{x}_i, \mathbf{n}_k) \tag{6.3}$$

par une procédure d'optimisation alternée. Chaque itération minimise le critère en calculant successivement les noyaux en fixant la partition donnée, puis la partition en fixant les noyaux.

6.1.3 Modèles de mélange et partitions

De nombreux autres algorithmes et heuristiques qui optimisent d'autres critères que celui de l'inertie intra-classes, ou qui simplement produisent une partition, existent. L'approche probabiliste permet de s'adapter à une grande variété de situations et généralise certaines techniques usuelles (nuées dynamiques, cartes de Kohonen).

L'approche probabiliste de la recherche de partitions fait l'hypothèse que l'ensemble des données $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ est la réalisation d'un échantillon de N variables aléatoires indépendantes de même loi f, prenant leurs valeurs dans $\mathbb{R}^d$. La connaissance de cette loi f doit permettre de séparer "naturellement" les N observations en k classes.

Les mélanges finis de densité sont les distributions de probabilité les plus utilisées dans ce contexte :

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}|\theta_k), \tag{6.4}$$

avec $\sum_{k=1}^{K} p_k = 1$. Ce genre de densité apparaît naturellement lorsque la population considérée est formée de plusieurs sous-populations qui ont des densités différentes. Ceci explique l'intérêt de ce modèle en classification. Les modèles de mélange, de loi de Gauss ou de Bernoulli, sont les modèles le plus souvent utilisés dans le contexte de la classification automatique.

Dans ce contexte, deux approches ont été proposées :

- l'approche mélange,
- l'approche classification.

La première approche considère effectivement que les individus observés sont les réalisation d'un mélange de densité, alors que la seconde approche traite chacune des sous-population de manière séparée.

Approche mélange

Si l'on considère que les vecteur forme observés $(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ sont des réalisations indépendantes d'une loi mélange :

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}|\theta_k),$$

alors la log-vraisemblance des paramètres $\Phi = (p_1, \cdots, p_{K-1}, \theta_1, \cdots, \theta_K)$ s'exprime comme :

$$L(\Phi; \mathbf{x}_1, \cdots, \mathbf{x}_N) = \sum_{i=1}^{N} \log\left(p_k f_k(x_i|\theta_k)\right).$$

En général l'algorithme EM (chapitre 3) est utilisé pour trouver des estimateurs du m.v. (maximum de vraisemblance). Dans le cadre de cette approche, une partition

des données peut être obtenue à partir des estimateur du m.v. en affectant chaque vecteur forme à la composante du mélange (donc la classe) la plus probable. La probabilité conditionnelle que $\mathbf{x}_i$ soit issu de la $k$composante est donnée par :

$$t_k(\mathbf{x}_i) = \frac{\hat{p}_k f_k(x_i|\hat{\theta}_k)}{\sum_{\ell=1}^{K} \hat{p}_\ell f_\ell(x_i|\hat{\theta}_\ell)}. \tag{6.5}$$

Approche classification

Une autre approche possible, dans une optique de partitionnement de l'échantillon, consiste à considérer directement la partition comme le paramètre inconnu. Les pionniers de cette approche sont **?**) et **?**). Dans ce contexte le problème à résoudre peut être formulé comme suit : étant donné un échantillon de taille N, $(\mathbf{x}_1, ..., \mathbf{x}_N)$, rechercher une partition dure $P = (\mathcal{C}_1, \cdots, \mathcal{C}_K)$, K étant supposé connu, telle que chaque classe $\mathcal{C}_k$ soit assimilable à un sous-échantillon suivant la loi $f_k(.|\theta_k)$.

Le critère considéré alors n'est plus la vraisemblance de l'échantillon, mais la vraisemblance classifiante, soit le produit des vraisemblance sur les classes. La log-vraisemblance s'écrit alors :

$$CML(\Phi, \mathbf{c}; \mathbf{x}_1, \cdots, \mathbf{x}_N) = \sum_{k=1}^{K} \sum_{i=1}^{n} c_{ik} \log \{f_k(\mathbf{x}_i|\theta_k\} \tag{6.6}$$

avec $\Phi = (\theta_1, \cdots, \theta_K)$ et $\mathbf{c} = \{c_{ik}\}$ une matrice de partition dure qui définit K classes (ou sous échantillons). Ce critère est la log-vraisemblance associée à K échantillons séparés de taille fixée.

La vraisemblance classifiante ne fait pas apparaître explicitement la notion de proportions entre les différentes sous-populations et tend en pratique à produire des partitions où les classes sont de tailles comparables. En fait le critère suppose implicitement que toutes les sous-populations sont de même taille. Cette limitation a incité **?**) à pénaliser la vraisemblance classifiante par un terme prenant en compte les proportions $(p_1, \cdots, p_K)$ des différents sous-échantillons :

$$CML'(\Phi', \mathbf{c}) = CML(\Phi, \mathbf{c}) + \sum_{k=1}^{K} n_k \log p_k \tag{6.7}$$

où $\Phi' = (\Phi, p_1, \cdots, p_K)$ et n_k est l'effectif de la $k$classe. Notons qu'en introduisant les variables c_{ik} dans le terme de pénalité, la vraisemblance classifiante pénalisée s'écrit

$$\begin{aligned} CML'(\Phi', \mathbf{c}) &= CML(\Phi, \mathbf{c}) + \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} \log p_k \\ &= \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} \log \{p_k f_k(\mathbf{x}_i|\theta_k)\}, \\ &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} \log P(\mathbf{x}_{i,k}, k; \Phi'), \end{aligned}$$

où $\mathbf{x}_{i,k}$ dénote un vecteur forme de la classe k. Ce dernier critère s'interprète comme la log-vraisemblance d'un échantillon aléatoire $\{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_N, c(\mathbf{x}_N))\}$ où à la différence d'un problème de classement les étiquettes $c(\mathbf{x}_i)$ n'ont pas été observées.

Ces deux critères peuvent être maximisés par une version classificatoire de l'algorithme EM : *Classification EM algorithm*. L'algorithme CEM a été proposé par **?**).

Une itération cet algorithme se décompose ainsi :

- **Étape E (estimation) :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ (cf. équation **??**) pour chaque $\mathbf{x}_i$.
- **Étape C (classification) :** Chaque $\mathbf{x}_i$ est affecté à la composante du mélange de plus forte probabilité a posteriori. Une partition P^{q+1} est donc définie caractérisée par la matrice $\mathbf{c} = \{c_{ik}\}$ avec $c_{ik} = 1$ si $k = arg\max_\ell t_\ell(x_i)^q$ et $c_{ik} = 0$ sinon.
- **Étape M (maximisation) :** Calcul des estimateurs du m.v. de Φ^{q+1} sur la base des sous-échantillons précisés par la matrice de classification dure $\mathbf{c}$.

L'algorithme CEM génère une suite $CML'(\Phi^q, \mathbf{c}^q)$ croissante qui atteint son maximum en un nombre fini d'itérations (**?**).

L'algorithme CEM est un algorithme très général de classification qui permet d'optimiser de nombreux critères de classification de type inertiels suivant les modèles gaussiens considérés. Prenons par exemple le modèle gaussien le moins contraint, pour lequel les classes sont de tailles différentes et possèdent une matrice de variance covariance quelconque, l'algorithme CEM maximise alors le critère de vraisemblance classifiante pénalisée. Si toutes les proportions sont fixées égales, le critère optimisé est alors simplement la vraisemblance classifiante. Un autre cas particulier intéressant est celui où les densités $f_k(.|\theta_k)$ du mélange sont des gaussiennes de vecteur moyenne $\boldsymbol{\mu}_k$ et de matrice de variance covariance $\boldsymbol{\Sigma}_k = \lambda \cdot I$, mélangés en proportions égales. En effet le critère optimisé est la somme des variances intra-classes,

$$\begin{aligned} CML(\Phi, \mathbf{c}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik} \log\left(2\pi \det|\boldsymbol{\Sigma}_k|\right)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^t(\lambda\cdot I)^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right) \\ &= -\frac{1}{2\lambda}\sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik}(\mathbf{x}-\boldsymbol{\mu}_k)^t(\mathbf{x}-\boldsymbol{\mu}_k) + Cst, \\ &= -\frac{1}{2\lambda} trace(_W) + Cst, \end{aligned}$$

et l'algorithme CEM, avec ce modèle, est exactement l'algorithme des centres mobiles présenté dans la section **??**.

Différentes études (**?**) ont montré que l'approche classification introduisait un biais dans l'estimation des paramètres. En effet, cette approche estime les paramètres du mélange sur la base des classes, alors que les classes sont disjointes et constituent en fait des échantillons tronqués des composantes du mélange. Ce phénomène a tendance

à surestimer les différences entre les moyennes, et à sous estimer les variances et les différences entre proportions. Ces inconvénients ne sont pas rédhibitoires si les classes sont bien séparées et les proportions du même ordre de grandeur.

Liens avec la classification floue

Dans le cadre de la reconnaissance des formes, l'algorithme EM pour les modèles de mélanges peut être interprété comme un algorithme d'optimisation alternée d'un certain critère (**?**, **?**).

Si les probabilités a posteriori $t_k(\mathbf{x}_i)$ sont considérées comme des variables notées c_{ik}, la log-vraisemblance $L(\Phi;\mathbf{x})$ devient une fonction du vecteur Φ et des c_{ik} que nous noterons :

$$L(\mathbf{c},\Phi) = \sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik}\log p_k f_k(\mathbf{x}_i|\theta_k) - \sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik}\log c_{ik}, \tag{6.8}$$

avec $\mathbf{c} = \{c_{ik} : 0 \leq c_{ik} \leq 1, \sum_{k=1}^{K} c_{ik} = 1, \sum_{i=1}^{N} c_{ik} > 0 (1 \leq i \leq N, 1 \leq k \leq K)\}$.

Considérons le problème qui consiste à maximiser $L(\mathbf{c},\Phi)$ par rapport aux variables Φ et $\mathbf{c}$. Il s'agit d'un problème classique d'optimisation sous contraintes. Une méthode d'optimisation possible consiste à séparer les variables en deux groupes et à optimiser le critère alternativement par rapport à un groupe en gardant fixe les valeurs des variable de l'autre groupe. Dans le cas du critère $L(\mathbf{c},\Phi)$, pour la $q$itération il est possible d'optimiser alternativement par rapport à $\mathbf{c}$ puis à Φ :

1. Maximisons $L(\mathbf{c},\Phi)$ par rapport à $\mathbf{c}$: le lagrangien s'écrit

$$\mathcal{L}(\mathbf{c}) = L(\mathbf{c},\Phi) + \sum_{i=1}^{N}\lambda_i\Big(\sum_{k=1}^{K}(c_{ik}-1)\Big), \tag{6.9}$$

où les λ_i sont les coefficients de Lagrange correspondant aux contraintes

$$\sum_{k=1}^{K} c_{ik} = 1.$$

Les conditions nécessaires d'optimalité amènent les équations suivantes :

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial c_{ik}} = \log\left(p_k f_k(\mathbf{x}_i|\theta_k)\right) - 1 - \log c_{ik} + \lambda_i = 0; \\ \sum_{k=1}^{K} c_{ik} = 1; \end{cases}$$

ce qui donne,

$$\begin{cases} c_{ik} = \exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i\right\}; \\ \sum_{k=1}^{K}\exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i\right\} = 1; \end{cases}$$

Ainsi les nouvelles valeurs des c_{ik} sont :

$$c_{ik} = \frac{p_k f_k(\mathbf{x}_i|\theta_k)}{f(\mathbf{x}_i)}. \tag{6.10}$$

2. La maximisation de $L(\mathbf{c}, \Phi)$ par rapport à Φ est équivalente l'étape M de l'algorithme EM.

Ces deux étapes qui visent à maximiser le critère $L(\mathbf{c}, \Phi)$ sont identiques aux deux étapes de l'algorithme EM appliqué à un mélange de distribution de probabilité.

Si l'on considère $\mathbf{c}$ comme une matrice de classification floue (elle en a toutes les caractéristiques), l'algorithme EM peut être interprété comme un algorithme de classification floue.

Remarquons que le critère optimisé s'écrit comme la somme de deux termes :

- Dans la terminologie utilisée en classification automatique, le premier est appelé "vraisemblance classifiante floue" (avec proportions libres). Plusieurs algorithmes de classification automatique existent qui visent à trouver la partition dure qui optimise la vraisemblance classifiante.
- Le second terme peut être considéré comme une entropie, ou encore une mesure de floue de la partition. Ce second terme est maximum si la partition obtenue est complètement floue et minimum (nul en l'occurence) $\mathbf{c}$ est une matrice de partition dure.

Au vu des remarques précédentes, l'algorithme EM peut être considéré comme un algorithme de classification flou qui optimise un critère de classification pénalisé par une entropie.

6.2 Hiérarchies

Définition 6.2 *$\mathcal{F}$ étant un ensemble fini, un ensemble H de parties non vides de $\mathcal{F}$ est une hiérarchie si :*

1. $\mathcal{F} \in H$
2. $\forall x \in \mathcal{F}$, $\{x\} \in H$
3. $\forall h, h' \in H$, $h \cap h' = \emptyset$ *ou* $h \subset h'$ *ou* $h' \subset h$

Une hiérarchie peut être vue comme un ensemble de partitions emboîtées. Graphiquement une hiérarchie est souvent représentée par une structure arborescente appelée dendogramme.

Exemple 6.2 Exemple de hiérarchie : en biologie les différente races d'animaux sont regroupées en espèces, qui sont elles même regroupées en grande famille...

△

Une hiérarchie peut être obtenue par deux types de méthodes, selon que l'arbre est construit en commençant par les feuilles ou bien la racine :

- la classification ascendante ("agglomérative") considère initialement chaque vecteur forme comme une classe. À chaque itération les deux classes les plus proches sont agrégées pour former une nouvelle classe. Le processus se termine naturellement lorsqu'il ne reste qu'une seule classe.

- la classification descendante ("divisive" en anglais), part d'une seule classe (l'ensemble des vecteurs forme) partage celle-ci en deux. L'opération est répétée à chaque itération jusqu'a ce toutes les classes obtenues contiennent un unique vecteur forme.

Il existe un parallèle intéressant entre la notion de distance ultramétrique et la notion de hiérarchie. Une distance utlramétrique δ vérifie toutes les propriétés qui définissent une distance classique et satisfait en plus l'inégalité

$$\delta(\mathbf{x}, \mathbf{z}) \leq \max\left(\delta(\mathbf{x}, \mathbf{z}), \delta(\mathbf{z}, \mathbf{y})\right),$$

plus forte que l'inégalité triangulaire. Lorsqu'on dispose d'une hiérarchie, on peut interpréter le nombre minimum d'emboitements nécessaires pour que deux vecteurs forme appartiennent à une même classe, comme une dissimilarité. Il est alors possible de montrer que cette dissimilarité est une distance ultramétrique. Ainsi, il est possible d'intepréter le problème de la classification hiérarchique comme la recherche d'une ultramétrique δ proche de d, la dissimilarité utilisée sur $\mathcal{F}$ l'ensemble à classer.

6.2.1 Classification ascendante hiérarchique

Le principe des algorithmes de classification hiérarchique ascendante est très simple :

Initialisation : chaque élément de $\mathcal{F}$ constitue une classe. Une "distance" D est calculée entre toutes les classes.

Tant que nombre de classes > 1

- regrouper les deux classes les plus proches au sens de la "distance" D,
- calcul des "distances" entre la nouvelle classe et les autres.

La "distance" D entre deux partie h et h' de $\mathcal{F}$, peut être définie de nombreuses manières à partir d'une mesure de dissimilarit'e d sur $\mathcal{F}$.

Critères d'agrégation

La "distance" D est couramment appelée critère d'aggrégation. Quatre variantes sont principalement utilisées :

- le critère du lien minimum ("single link") :

$$D(h, h') = \min\left[d(\mathbf{x}, \mathbf{y}) / \mathbf{x} \in h \; et \; \mathbf{y} \in h'\right],$$

- le critère du lien maximum ("complete link") :

$$D(h, h') = \max\left[d(\mathbf{x}, \mathbf{y})/\mathbf{x} \in h \; et \; \mathbf{y} \in h'\right],$$

- le critère de la distance moyenne ("group average") :

$$D(h, h') = \frac{\sum_{i=1}^{n_h} \sum_{j=1}^{n_{h'}} d(\mathbf{x}_i, \mathbf{x}_j)}{n_h \cdot n_{h'}},$$

- le critère de **?**)

$$D(h, h') = \frac{n_h \cdot n_{h'}}{n_h + n_{h'}} \|\mathbf{m}_h - \mathbf{m}_{h'}\|^2.$$

Critère d'inertie intra classe et méthode de Ward

Lorsque l'on dispose d'une partition en K classe, le critère d'inertie intra-classe mesure son homogénéité :

$$\begin{aligned} I_W &= trace(_W), \\ &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\mathbf{x}_{ik} - \mathbf{m}_k)^t (\mathbf{x}_{ik} - \mathbf{m}_k). \end{aligned}$$

Considérons deux partitions :

- $P = (\mathcal{C}_1, \cdots, \mathcal{C}_K)$,
- et P', la partition obtenue en fusionnant les classes $\mathcal{C}_k$ et $\mathcal{C}_\ell$.

On peut montrer que la différence entre l'inertie des deux partitions est égale au critère d'aggrégation de Ward :

$$I_{W'} - I_W = \frac{n_k \cdot n_\ell}{n_k + n_\ell} \|\mathbf{m}_k - \mathbf{m}_\ell\|^2.$$

Ainsi,à chaque étape de l'algorithme de Ward choisit une nouvelle partition qui limite l'augmentation de l'inertie intra-classe. Notons que cette propriété ne garantie pas l'optimisation globale du critère.

6.2.2 Classification descendante hiérarchique

La classification descendante est beaucoup moins populaire que les méthodes décrites précédemment. En théorie, la première étape d'une méthode descendante doit comparer les $2^{N-1} - 1$ partitions possibles des N vecteurs forme, en deux classes. Pour éviter des calculs impossibles, une solution consiste à appliquer une méthode de partitionnement pour obtenir les deux classes. En répétant ce processus récursivement sur chaque classe obtenue, il en résulte une hiérarchie.

Chapter 7

Données spatiales et reconnaissance des formes

Dans ce chapitre nous considérons des vecteurs formes qui possèdent une localisation spatiale (par exemple la position d'un pixel dans une image). Les variables indiquant la position "géographique" revêtent souvent une importance particulière par rapport aux autres variables et doivent être traitées de manière distincte.

Dans le contexte d'un problème de classement ou bien de classification, il semble naturel de vouloir visualiser la partition dans l'espace géographique. Si des techniques classiques de reconnaissance des formes sont utilisées pour discriminer ou classifier, la partition obtenue sera en générale spatialement très morcelée. Pour éviter ce morcellement, il faut considérer l'information spatiale des données, c'est-à-dire le fait que deux vecteurs forme spatialement proches tendent à appartenir à la même classe.

La prise en compte des dépendances spatiales peut s'effectuer au niveau :

- du prétraitement,
- du classement (ou de la classification).

Ce chapitre présente quelques exemples de prétraitements possibles, et insiste surtout sur les méthodes de classification ou de classement qui intégrènt la notion de dépendance spatiale. Comment intégrer l'information spatiale dans un algorithme de reconnaissance des formes ? Il est possible de recourir au bon sens, si bien partagé, pour modifier certains algorithmes classiques, mais on peut aussi choisir de partir d'un modèle statistique raisonnable incorporant la notion de dépendance spatiale.

Les statistiques spatiales offrent une grande variété de choix de modèles. Ces statistiques trouvent leurs applications dans tous les domaines où les données à traiter sont localisées spatialement comme en astronomie, exploitation minière, écologie, géographie et archéologie... Cette branche de la statistique vise à répondre à des questions aussi diverses que :

- Comment résumer un ensemble de données spatiales par des statistiques et des graphiques pertinents ?
- Est-ce que les arbres d'une forêt sont répartis au "hasard", ou bien existe-t-il une structure sous-jacente ?
- Tel modèle statistique explique-t-il mieux que tel autre la répartition spatiale des données observées ?
- Quelle température fait-il à Paris, connaissant la température de certaines villes voisines ?

Suivant le type de données considérées, les intérêts et les méthodes sont différents (**?**). Si la localisation des individus (individu au sens de l'analyse des données) et les distances entre individus sont le phénomène de première importance, les données analysées seront des points dans l'espace. Par contre, si des mesures localisées spatialement sont la matière première de l'analyse statistique, les individus étudiés seront des vecteurs localisés.

Dans tous les cas, l'ensemble des données est considéré comme la réalisation d'un processus stochastique :

Définition 7.1 *Un processus stochastique* $\{\mathbf{X}_t\}$ *est une suite de v.a. indicées sur un sous-ensemble de* $\mathbb{R}^d$.

Souvent l'indice représente le temps. Dans le cadre des statistiques spatiales, l'indice figure les coordonnées dans l'espace (la plupart du temps le plan) et la variable aléatoire $\mathbf{X}_t$ peut signifier l'absence ($\mathbf{X}_t = 0$) ou la présence ($X_t = 1$) d'un point en t, la température à l'endroit t...

Le type de données détermine la classe de modèles statistiques pris en compte. Ainsi plusieurs classes de processus peuvent être distinguées :

- Les processus stochastiques générateurs de points (Stochastic Point Processes en anglais), qui modélisent la répartion spatiale de points.

  Exemple 7.1 Dans le cas où l'on s'intéresse uniquement à la répartition spatiale d'un ensemble d'individus (des arbres, des villes...) à l'intérieur d'une zone géographique définie, les processus stochastiques générateurs de points sont des modèles statistiques adaptés.

  Ainsi, la première étape de l'analyse consiste à déterminer si les points sont répartis au hasard ou forment une structure plus complexe. Ce problème relève de la théorie des tests d'hypothèse et fait intervenir le processus de Poisson :

  H_0 : Les données sont la réalisation d'un processus de Poisson
  H_1 : Les données ne sont pas réparties au "hasard"

  △

- Les processus générateurs de variables régionalisées, qui modélisent des phénomènes spatialement continus (par exemple, la température en France). Ce genre de processus sert essentiellement dans la théorie de l'interpolation spatiale.

- Les séries spatiales qui sont une sorte de généralisation des série temporelles, qui prennent en compte des données localisées, en un certain nombre de sites (Par exemple, la hauteur des arbres dans une forêt).

Une présentation détaillée des processus stochastiques spatiaux peut être trouvée dans **?**) et **?**).

Dans le cadre de la discrimination et de la classification d'un ensemble de vecteurs forme, seules les variables régionalisée et les séries spatiales fournissent des modèles exploitables.

7.1 Modifications des données

7.1.1 Utilisation des variables spatiales

Traiter les variables de positions spatiales au même titre que les autres variables décrivant les sites, semble être une idée naturelle. Les coordonnées spatiales peuvent être pondérées pour contrôler la quantité d'information spatiale qui sera prise en compte par l'algorithme de classement ou de classification automatique utilisé. Cette idée remonte à **?**) et a aussi été utilisée en segmentation d'image par **?**). D'après **?**) ce genre d'approche souffre du même défaut que la précédente : elle tend à séparer dans des classes différentes deux sites qui sont très similaires mais qui sont éloignés géographiquement.

7.1.2 Transformation des variables

Au lieu de travailler sur le tableau individus/variables initial, il est possible de commencer par une phase de prétraitement. Cette phase de prétraitement a pour but d'extraire de nouvelles variables qui contiennent l'information spatiale.

Exemple 7.2 Une possibilité consiste à définir une taille de fenêtre géographique et à remplacer les variables initiales d'un vecteur forme par une combinaison linéaire des variables de ce vecteur forme et de ses voisins géographiques (c.-à-.d., à l'intérieur de la fenêtre centrée sur cet individu).

△

7.1.3 Utilisation de la matrice des distances spatiales

Si les données initiales prennent la forme d'un tableau de distances individu/individu, on peut transformer cette matrice pour intégrer l'information spatiale.

?) proposent d'utiliser les distances géographiques g_{ij} entre les sites i et j pour modifier les distances ou dissimilarités d_{ij} calculées avec les variables non géographiques. Il résulte de cette opération que la méthode de classification utilisée, part d'une nouvelle matrice de dissimilarité $D^* = \{d^*_{ij}\}$ qui mélange informations géographiques et non géographiques. Les algorithmes de classification usuels (non contraints) partitionnent les données.

Exemple 7.3 (**?**) Partant d'une matrice de dissimilarités $D = \{d_{ij}\}$, la démarche suivante fournit une partition des données qui évite un trop grand morcellement géographique sans toutefois produire des classes totalement connexes :

- Modification de la matrice des dissimilarités :

$$d^*_{ij} = d_{ij} \cdot [1 - \exp(-g_{ij}/W)]$$

  avec W un coefficient arbitraire. Plus W est grand, plus la nouvelle matrice des dissimilarités D^* est influencée par les distances géographiques et moins la partition sera fragmentée.

- Transformation de la matrice D^* en un tableau individus/variables par une analyse factorielle.

- Partitionnement du nouveau tableau de données par l'algorithme des k-means.

△

7.2 Classification hiérarchique contrainte

Au cours d'un processus de classification agglomératif, il est possible de restreindre les regroupement aux entités qui sont géographiquement voisines. Les contraintes de contiguïté sont alors respectées de façon absolue et les classes produites sont connexes, c'est à dire qu'une classe forme une seule région géographique, un seul bloc (**?**, **?**, **?**). Ces procédures de classification rangent dans des classes séparées deux sites qui sont spatialement très éloignés même s'ils sont très similaires au niveau des variables non géographiques. L'information spatiale joue alors un rôle prépondérant. Ce genre d'approche n'autorise pas la variation de "la quantité d'information spatiale" utilisée dans le processus de classification.

Produire des classes géographiquement connexes exige de définir au préalable quel individu est spatialement voisin de quel autre. La définition des rapports de voisinage est équivalente à la construction d'un graphe non orienté où chaque nœud est un élément de l'ensemble des données et chaque arête figure une relation de voisinage.

Une classification automatique avec contrainte de contiguïté absolue peut être partagée en deux étapes:

1. La définition d'un graphe de voisinage. Ceci peut être réalisé par une triangulation de Delaunay, par un graphe de Gabriel, par une grille...
2. La classification avec contraintes. Il est possible de modifier certains algorithmes classiques pour respecter les contraintes résumées par le graphe.

Exemple 7.4 (**?**) La classification hiérarchique ascendante est une méthode de classification simple et utilisable pour des ensembles de données de taille raisonnable (moins de 10000 individus). L'ajout de contraintes spatiales peut se faire de manière naturelle :

- **Initialisation :** calcul du graphe de voisinage et des distances entre individus deux à deux (Chaque individu est considéré comme une classe).
- **Itérer :** tant que le nombre de classes est supérieur à un :
  - regrouper les deux classes qui sont les plus proches au sens d'un certain critère d'agrégation parmi les classes voisines au sens du graphe de voisinage,
  - recalculer la matrice des distances et le graphe de voisinage entre les nouvelles classes.

Le fait de chercher seulement parmi les voisins géographiques quelles sont les classes les plus proches réduit de beaucoup l'espace de recherche et accélère la procédure.

△

7.3 Discrimination linéaire et corrélation spatiale

Pour prendre en compte la continuité spatiale d'une image, dans le contexte de la discrimination, **?**) propose de considérer des vecteurs formes augmentés $(\mathbf{x}_1^+, \cdots, \mathbf{x}_N^+)$ à la place des vecteurs forme initiaux $(\mathbf{x}, \cdots, \mathbf{x}_N)$. La présentation plus générale de **?**) définie le vecteur forme augmenté $\mathbf{x}_i^+$ comme

$$\mathbf{x}_i^+ = (\mathbf{x}_i^t, \mathbf{x}_{G_i}^t)^t,$$

avec $\mathbf{x}_{G_i}$ le vecteur forme constitué par les s vecteurs voisins de $\mathbf{x}_i$:

$$\mathbf{x}_{G_i} = (\mathbf{x}_{i1}, \cdots, \mathbf{x}_{is})^t.$$

Dans le but de rendre possible l'expression d'une fonction discriminante linéaire, les hypothèses suivantes sont faites :

- chaque vecteur forme observé $\mathbf{x}_i$ est la somme d'un vecteur moyenne et d'un bruit gaussien ϵ_i :

$$\mathbf{x}_i = \sum_{k=1}^{K} c_{ik} \boldsymbol{\mu}_k + \epsilon_i,$$

- le bruit $\epsilon_1, \cdots, \epsilon_N$ est multi-gaussien de moyenne nulle (identique pour toute les classes).
- on suppose une corrélation spatiale intrinsèque :

$$cov\left[\mathbf{X}_i, \mathbf{X}_j\right] = \rho(d)\mathbf{\Sigma},$$

où d est la distance séparant $\mathbf{X}_i$ et $\mathbf{X}_j$. Notons que $\rho(0) = 1$.

- on suppose une très forte continuité spatiale : si un pixel $\mathbf{x}_i$ appartient à la classe k alors ses voisins appartiennent à la même classe avec une probabilité proche de 1.

Ainsi sous toutes ces hypothèses on peut déterminer la forme des vecteur moyennes et matrice de covariance des vecteurs formes augmentés :

- $$\boldsymbol{\mu_k^+ = I\!E\left[\mathbf{X_i^+} | \mathbf{X_i} \in \mathcal{C}_k\right] = \mathbf{1}_{s+1} \otimes \boldsymbol{\mu}_k}$$

où 1_{s+1} est est vecteur colonne de dimension $s+1$ (s est le nombre de voisins)

- $$\mathbf{\Sigma}^+ = \mathbf{C} \otimes \mathbf{\Sigma}$$

où $\mathbf{C}$ est la matrice de corrélation spatiale entre le composante de $\mathbf{x}_i^+$. Sa taille est donc de $(s+1) \times (s+1)$.

Pour construire les fonctions discriminantes, on suppose que les vecteurs augmententés $\mathbf{x}_i^+$ d'une classe k suivent une loi multi-gaussienne de vecteur moyenne $\boldsymbol{\mu}_k^+$ et de matrice de variance $\mathbf{\Sigma}^+$. Cette hypothèse Gaussienne homoscédastique amène à considérer des fonctions discriminantes linéaires.

7.4 Approche globale

Pour traiter des données non spatiales, les modèles statistiques utilisés dans le contexte de la discrimination ou de la classification supposent que les vecteurs forme sont indépendants : la loi jointe d'un ensemble de vecteurs forme donné est le produit des densité de chaque vecteur forme.

Une manière d'aborder le problème, en introduisant la notion de dépendance spatiale, consiste à faire des hypothèses directement sur la forme de la loi jointe. Dans ce sens ce type d'approche peut être qualifié de global.

La modélisation statistique globale des images pour la segmentation suppose l'existence de deux champs aléatoires. L'image observée, $\mathbf{x}$, est la réalisation d'un premier champ aléatoire $\mathbf{X} = \{\mathbf{X}_s, s \in S\}$ et l'image segmentée, $\mathbf{c}$, est la réalisation d'un second champ $\mathbf{C} = \{\mathbf{C}_s, s \in S\}$ (avec S, l'ensemble des pixels). Les variables aléatoires $\mathbf{X}_s$ prennent leur valeur dans $I\!R^d$ et les $\mathbf{C}_s$ dans un ensemble fini

$\Omega = \{\omega_1, \cdots, \omega_K\}$ avec K le nombre de classes. Le modèle considère que $\mathbf{X}$ est une observation bruitée de $\mathbf{C}$. Ainsi une relation existe entre les deux champs :

$$\mathbf{X} = R(\mathbf{C}, \mathbf{N}), \tag{7.1}$$

où $\mathbf{N}$ est le bruit.

Le problème de la segmentation supervisé (classement) consiste à trouver un estimateur $\hat{\mathbf{c}}$ de $\mathbf{c}$ lorsque l'on dispose d'exemple d'images segmentées ou bien de zones segmentées dans une image. La segmentation non supervisée est concernée par le problème plus délicat d'estimation $\hat{\mathbf{c}}$ en l'absence d'exemple de segmentation.

Comme dans le cas des données non spatiales la théorie bayésienne de la décision permet de définir un cadre formel à ces problèmes. Notons

- $P(\mathbf{X} = \mathbf{x}|\mathbf{C} = \mathbf{c})$ la loi des données conditionnelle à la connaissance du champ $\mathbf{c}$,
- la loi *a priori* $P(\mathbf{C} = \mathbf{c})$ sur l'image segmentée,
- le coût $L(\mathbf{c}, \hat{\mathbf{c}}(\mathbf{x}))$ associé à la décision $\hat{\mathbf{c}}(\mathbf{x})$ sachant que $\mathbf{C} = \mathbf{c}$.

Notons que la distribution de probabilité $P(\mathbf{X}|\mathbf{C})$ est déterminée par la relation qui existe entre les champs $\mathbf{X}$ et $\mathbf{C}$ (Equation **??**). La distribution a posteriori peut être exprimée par le théorème de Bayes :

$$P(\mathbf{C} = \mathbf{c}|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{C} = \mathbf{c})P(\mathbf{X} = \mathbf{x}|\mathbf{C} = \mathbf{c})}{P(\mathbf{X} = \mathbf{x})}$$

La stratégie bayésienne consiste alors à prendre la décision qui minimise le risque conditionnel :

$$\hat{\mathbf{c}} = arg \min_{\mathbf{C}} R(\mathbf{c}|\mathbf{x})$$

avec

$$R(\mathbf{c}|\mathbf{x}) = I\!E\left[L(\mathbf{Z}, \mathbf{c})|\mathbf{x}\right] = \sum_{\mathbf{Z}} L(\mathbf{z}, \mathbf{c})P(\mathbf{C} = \mathbf{z}|\mathbf{X} = \mathbf{x}),$$

où $L(\mathbf{z}, \mathbf{c})$ est le coût de dire que l'image segmentée est $\mathbf{c}$ lorsque l'image segmentée est en fait $\mathbf{z}$. Deux fonctions de coût sont couramment utilisées en analyse d'image :

- $L(\mathbf{z}, \mathbf{c}) = I\!I_{\{\mathbf{c} \neq \mathbf{z}\}}$, c'est le coût "0–1" qui vaut 0 pour la bonne décision et 1 pour une mauvaise décision. Dans ce cas l'estimateur de $\mathbf{c}$ est le maximum a posteriori (MAP) :
$$\hat{\mathbf{c}} = arg \max_{\mathbf{C}} P(\mathbf{C} = \mathbf{c}|\mathbf{x})$$
- $L(\mathbf{z}, \mathbf{c}) = \sum_{s \in S} I\!I_{\{\mathbf{c}_s \neq \mathbf{z}_s\}}$, qui considère non plus l'image dans son intégralité mais le nombre de pixels bien classés. Dans ce cas l'estimateur est celui qui maximise les probabilités marginales a posteriori (MPM) (**?**):
$$\hat{\mathbf{c}_s} = arg \max_{\mathbf{C}_s} P(\mathbf{C}_s = \mathbf{c}_s|\mathbf{x}),\ \forall s \in S.$$

Cette approche globale du problème de segmentation partage des problèmes communs avec la reconnaissance statistique des formes classique, mais se distingue de plusieurs manières. Ainsi comme en reconnaissance statistique des formes, il est nécessaire de répondre aux questions suivantes :

1. Quel modèle relatif aux données observées adopter ($P(\mathbf{X}|\mathbf{C})$) ?

2. Comment estimer les paramètres du modèle choisi ?

Les problèmes particuliers soulevés par cette approche sont les suivants :

- Comment modéliser les connaissances a priori sur la structure de l'image segmentée ($P(\mathbf{C})$) ? En effet, dans le cas de données non spatiales cette probabilité est simplement une loi multinomiale, ce qui ne semble pas du tout adapté pour la prise en compte des dépendances spatiales.

- Comment trouver l'estimateur de l'image segmentée qui minimise la fonction de coût choisie ? Concernant les données non spatiales, lorsque les paramètres de la loi *a posteriori* sont déterminés, l'optimisation du risque conditionnel est direct (pour le coût {0,1}), ce qui est loin d'être le cas pour l'approche globale décrite précédemment.

Notons aussi que à la différence de la reconnaissance statistique des formes classique qui raisonne sur un échantillon de taille N, l'approche globale dispose le plus souvent d'une seule image (ou jeu de données) et donc d'un échantillon de taille 1.

Les méthodes globales de segmentation résolvent souvent le problème du choix de modèle en utilisant les champs de Markov utilisés en statistique spatiale.

7.5 Séries spatiales : modèles markoviens

Dans le cas où le processus $\{\mathbf{X}_t\}$ est défini en un certain nombre de sites spécifiques, et prend ses valeurs sur un ensemble discret ou continu, il existe des modèles, qui possèdent beaucoup de points communs avec les séries temporelles.

Ces modèles, comme dans le cas des processus générateurs de points, sont des alternatives à l'hypothèse d'absence d'autocorrélation spatiale (**?**).

La classe de modèles la plus répandue dans ce contexte est représentée par les champs aléatoires de Markov.

Champs aléatoires de Markov

Les chaînes de Markov sont des processus stochastiques les plus simples pour tenir compte de la non indépendance des v.a. $\mathbf{X}_n$ par rapport à un indice discret n :

Définition 7.2 *Un processus stochastique* $\{\mathbf{X}^n : n = 1, 2, ...\}$ *prenant ses valeurs dans un espace fini est une chaîne de Markov si la réalisation de* $\mathbf{X}^n$ *sachant toutes les réalisations passées ne dépend que de la dernière valeur prise :*

$$P(\mathbf{X}^n = \mathbf{x}^n | \mathbf{X}^{n-1} = \mathbf{x}^{n-1}, ..., \mathbf{X}^1 = \mathbf{x}^1) = P(\mathbf{X}^n = \mathbf{x}^n | \mathbf{X}^{n-1} = \mathbf{x}^{n-1}). \tag{7.2}$$

Ce concept de dépendance markovienne peut être étendu de bien des manières. Les champs de Markov sont une extension de la notion de dépendance markovienne pour des processus stochastiques dont l'indice appartient à un espace multidimensionel et plus seulement à un sous-ensemble de $\mathbb{R}$. Deux cas sont à distinguer :

- les champs de Markov dont l'indice varie de façon continue ;
- les champs de Markov dont l'indice est discret.

Les premiers trouvent leur domaine d'application en physique théorique et les seconds servent entre autre de modèles pour les statistiques ayant un caractère spatial. Seul le second cas sera examiné dans ce document.

Si l'indice n'appartient pas à un sous ensemble de $\mathbb{R}$ mais à un sous ensemble de $\mathbb{R}^d$, les notions de passé et de futur par rapport à un indice t ne tiennent plus, et il faut recourir au concept plus général de voisinage.

Définition 7.3 *(?) Soit* $S = \{s_1, s_2, ..., s_N\}$ *un ensemble d'indices (dans un contexte de modélisation spatiale, l'indice représente les coordonnées d'un site).* $G = \{G_s, s \in S\}$ *un ensemble de parties de* S *est un système de voisinage pour* S *si, et seulement si,* $\forall r, s \in S$,

1. $s \notin G_s$,
2. $s \in G_r \Leftrightarrow r \in G_s$.

Notons que $\{S, G\}$ est un graphe.

Un autre concept utile lié à la notion de système de voisinage est celui de clique :

Définition 7.4 *Soit* G *un système de voisinage sur un ensemble de sites* S*, une clique* c *est un sous-ensemble de* S *tel que tous les éléments de* c *soient voisins les uns des autres au sens de* G*.*

L'ensemble des indices (des sites) d'un graphe de voisinage forme un réseau. On distingue les réseaux à mailles régulières et les réseaux à mailles irrégulières. Les premiers sont utilisés pour modéliser la distribution d'une population (végétale, animale...) échantillonnée de manière très régulière lors d'une expérience. Les seconds sont utilisés pour décrire la répartition naturelle d'une population.

Exemple 7.5 Toutes les communes composant un département sont caractérisées par des nombres liés à leur activité agricole. L'activité agricole ne semble pas indépendante de la localisation d'une commune et il semble judicieux de modéliser la répartition spatiale de cette activité par un champ de Markov. L'ensemble S des indices peut être choisi comme les coordonnées du centre de la commune et deux communes sont considérées comme voisines si elles partagent une frontière commune. Les mailles de ce réseau sont irrégulières.

△

Lorsque les relations de voisinage ne sont pas définies de manière explicite et si les sites ne sont pas répartis régulièrement, il faut définir précisément la notion de voisinage avant de pouvoir recourir à une modélisation markovienne. Une solution possible consiste à dessiner une tesselation de Voronoï, et dire que deux sites sont voisins si leurs polygones de Voronoï respectifs partagent un côté commun.

Définition 7.5 *Soit S un ensemble d'indices muni d'un système de voisinage G et $\mathbf{X} = \{\mathbf{X}_s, s \in S\}$ une famille de variables aléatoires prenant leurs valeurs sur Ω. $\mathbf{X}$ est un champ de Markov par rapport à G si :*

1. $P(\mathbf{X} = \mathbf{x}) > O, \forall \mathbf{x} \in \Omega$;
2. $P(\mathbf{X}_s = \mathbf{x}_s | \mathbf{X}_r, r \neq s) = P(\mathbf{X}_s = \mathbf{x}_s | \mathbf{X}_r, r \in G_s), \forall s \in S.$

Cette définition affirme que l'état d'un site ne dépend que des voisins immédiats de ce site, mais elle n'est pas directement utilisable en pratique pour définir un champ de Markov sans la connaissance du théorème d'Hammersley-Clifford démontré en 1971 :

Théorème 7.1 *Soit $\mathbf{X} = \{\mathbf{X}_s, s \in S\}$ un champ de Markov sur un réseau S de n sites, muni d'un système de voisinage. La distribution de probabilité du champ $\mathbf{X}$ est une distribution de Gibbs:*

$$\pi(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z(T)} e^{-E(\mathbf{X})/kT},$$

- *où $Z(T)$ est une constante de normalisation, appelée constante de partition. Dans le cas où $\mathbf{X}$ prend ses valeurs sur un ensemble fini :*

$$Z(T) = \sum_{\mathbf{X}} e^{-E(\mathbf{X})/kT}.$$

- *et surtout où la fonction d'énergie E est de la forme:*

$$E(\mathbf{x}) = \sum_{1 \leq i \leq n} x_i G_i(x_i) + \sum \sum_{1 \leq i < j \leq n} x_i x_j G_{i,j}(x_i, x_j) + \cdots + x_1 x_2 \cdots x_n G_{1,2,\cdots,n}(x_1 x_2 \cdots x_n),$$

telle que pour tout $1 \leq i < j \cdots < s \leq n$, la fonction $G_{i,j,\cdots,s}$ peut être non nulle si et seulement si les sites $i, j, \cdots, s$ forment une clique.

Notons que la constante de partition est dans la grande majorité des cas incalculable (c'est une somme sur toutes les images possibles), et que la fonction d'énergie peut s'exprimer de manière plus simple comme une somme de fonction sur les cliques $\mathcal{C}$:

$$E(\mathbf{x}) = \sum_{\mathcal{C}} V_{\mathcal{C}}(\mathbf{x}).$$

7.5.1 Un modèle binaire

Comme le fait observer **?**), dans le cas où les variables de chaque site sont binaires, les fonctions G peuvent être remplacées par de simple paramètres sans perte de généralité. Si on se limite aux cliques de un et deux sites, la fonction d'énergie considérée aura la forme suivante:

$$E(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i x_i + \sum_{i=1}^{N} \sum_{j \in G_i, i<j} \beta_{i,j} x_i x_j \tag{7.3}$$

et la probabilité conditionnelle d'avoir $X_i = x_i$ sachant la réalisation de tous les voisins sera simplement:

$$P(X_i = x_i | X_j, j \neq i) = \frac{\exp\left(x_i(\alpha_i + \sum \beta_{i,j} x_j)\right)}{1 + \exp\left(\alpha_i + \sum \beta_{i,j} x_j\right)} \tag{7.4}$$

Exemple 7.6 (**?**) Le modèle de champ de Markov le plus connu trouve son origine en mécanique statistique. Il s'agit du modèle d'Ising inventé en 1925 pour expliquer certaines propriétés des ferromagnétiques. Les variables X_s (qui représentent la valeur du 'spin' d'un atome) peuvent prendre deux valeurs $+1$ ou -1, et sont associées aux sites d'un réseau hypercubique S muni d'un système de voisinage. À l'équilibre, la probabilité que le système soit dans une configuration $\mathbf{x}$ est une distribution de Gibbs de fonction d'énergie :

$$E(\mathbf{x}) = \alpha \sum_{s \in S} x_s + \beta \sum_{r,s \in S/r \text{ et } s \text{ voisins}} x_s x_r, \tag{7.5}$$

avec α et β des paramètres mesurant respectivement le champ magnétique extérieur et les forces de liaison. Lorsque $\alpha = 0$ (pas de champ extérieur) et que la température est grande toutes les configurations deviennent équiprobables et lorsque la température est basse deux configurations dominent : celle où tous les spins valent $+1$ et celle où tous les spins valent -1. A basse température le système reste piégé dans l'un des deux états et met très longtemps à en sortir. Ceci explique le phénomène d'aimantation rémanente.

△

7.5.2 Le modèle de Strauss (1977)

Le modèle de Strauss peut être considéré comme une généralisation du modèle d'Ising, dans le cas où les variables prennent des valeurs discrètes. Dans le cas isotrope, la distribution de Gibbs est définie par la fonction d'énergie :

$$E(\mathbf{x}) = \beta \sum_{r,s \in S/r \ et \ s \ voisins} I\!I_{\{\mathbf{x}_s = \mathbf{x}_r\}}. \tag{7.6}$$

Cette fonction d'énergie compte le nombre de paires de sites voisins qui ont la même valeur. Elle est maximum si les variables de tous les sites prennent une même valeur.

7.5.3 Des modèles gaussiens

Dans de nombreux cas, il est raisonnable de modéliser la distribution jointe des sites (ou plutôt d'une certaine variable en chaque site) par une loi normale multidimensionnelle. Dans cette optique, deux approches sont possibles :

- l'approche simultanée dite SAR (Simultaneous Autoregression) ;
- l'approche conditionnelle dite CAR (Conditional Autoregression).

La première solution définit le processus par N équations auto-régressives simultanées :

$$X_i = \mu_i + \sum \beta_{i,j}(X_j - \mu_j) + \epsilon_i, \tag{7.7}$$

où $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (**?**). Cette définition correspond à la distribution de probabilité suivante :

$$P(\mathbf{X} = \mathbf{x}) = (2\pi\sigma^2)^{\frac{-1}{2}N} det(\mathbf{B}) exp\{\frac{-1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{B}^T \mathbf{B}(\mathbf{x} - \boldsymbol{\mu})\}, \tag{7.8}$$

où $\boldsymbol{\mu}$ est un vecteur de taille N contenant les moyennes μ_i de tous les sites, et $\mathbf{B}$ une matrice $N \times N$ inversible qui contient des 1 sur la diagonale et les terme $-\beta_{i,j}$ partout ailleurs.

La deuxième approche définit le modèle de manière conditionnelle,

$$I\!E(\mathbf{X}_i | X_j = x_j, j \neq i) = \mu_i + \sum \beta_{i,j}(x_j - \mu_j).$$

et

$$var(\mathbf{X}_i | X_j = x_j, j \neq i) = \sigma^2.$$

Dans ce cas la densité du champ gaussien s'écrit :

$$P(\mathbf{X} = \mathbf{x}) = (2\pi\sigma^2)^{\frac{-1}{2}N} det(\mathbf{B})^{\frac{1}{2}} exp\{\frac{-1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{B}(\mathbf{x} - \boldsymbol{\mu})\}. \tag{7.9}$$

7.5.4 Application à la segmentation

Si le modèle de l'image est un champ de Markov, alors la distribution des données, les probabilités a priori et a posteriori sont des distributions de Gibbs définies de la façon suivante :

$$\begin{aligned} P(\mathbf{x}|\mathbf{c}) &\propto \exp -U^r(\mathbf{x},\mathbf{c},\Phi) \\ P(\mathbf{c}) &\propto \exp -U^a(\mathbf{c},\beta) \\ P(\mathbf{c}|\mathbf{x}) &\propto \exp\{-U^a(\mathbf{c},\beta) - U^r(\mathbf{x},\mathbf{c},\Phi)\} \end{aligned}$$

où Φ et β sont les paramètres des distributions. L'énergie U^a est relative aux informations a priori à propos de l'image segmentée $\mathbf{c}$. Plus $\mathbf{c}$ respecte les informations a priori et plus U^a est petite. L'énergie U^r est dite énergie de rappel aux données, c'est par son intermédiaire que la relation entre les champs $\mathbf{X}$ et $\mathbf{C}$ est modélisée.

Exemple 7.7 Au cours de l'étude d'une céramique qui contient des grains de carbure de silicium, on veut déterminer le pourcentage de carbure de silicium contenu dans le matériau. Une méthode possible est de prendre une photo d'une surface de ce matériau puis déterminer quelle surface est couverte par les grains de carbure de silicium. Le rapport entre cette surface et la surface totale donne une approximation du pourcentage cherché. La photo est numérisée et l'image résultante est codée en 256 niveaux de gris. La première étape de l'analyse consiste donc à segmenter l'image en deux classes, c'est-à-dire distinguer les pixels qui représentent le carbure de silicium du reste.

Formellement, on peut noter que $\mathbf{C}_s$ (la v.a. indiquant la classe) prend ses valeurs dans $\{0,1\} \times \{0,1\}$ ($\mathbf{C}_s = [01]^t$ indique que le site s appartient à la première classe). Deux pixels voisins ont plus de chance d'appartenir à la même classe que deux pixels quelconques. Cette information a priori peut être modélisée par le modèle d'Ising qui est une distribution de Gibbs de fonction d'énergie :

$$U^a(\mathbf{c}) = -\beta \sum_{r,s \in S/r \text{ et } s \text{ voisins}} \mathbf{c}_s \cdot \mathbf{c}_r. \tag{7.10}$$

Si l'on considère que l'image observée est une dégradation de l'image segmentée telle qu'en chaque pixel x_s de la classe k le bruit est gaussien de moyenne $\boldsymbol{\mu}_k$ et de variance $1/\Phi$, l'énergie de rappel aux données est :

$$U^r(\mathbf{x},\mathbf{c},\beta) = -\Phi \sum_{s \in S} (x_s - \sum_{k=1}^{K} c_{sk}\boldsymbol{\mu}_k)^2 \tag{7.11}$$

△

Les informations sur les relations entre les champs $\mathbf{X}$ et $\mathbf{C}$ et les a priori sur la forme du champ $\mathbf{C}$ peuvent être associée à des énergies. L'énergie de la distribution a posteriori est dans ce cas la somme de toutes ces énergies :

$$U = \sum U_i = U^a + U^r \tag{7.12}$$

Si la traduction des connaissances *a priori* dans une formulation markovienne semble assez aisée, l'estimation des paramètres du modèle et la segmentation de l'image constituent des problèmes délicats.

7.5.5 Minimisation du risque conditionnel et simulation

Dans cette section, les paramètres des modèles (loi a priori et loi sur les données) sont supposés connus. Dans ce contexte, l'approche bayésienne du problème de segmentation cherche à trouver une image segmentée qui minimise le risque conditionnel. Si la fonction de coût $\{0, 1\}$ est utilisée, cela revient à chercher l'image la plus probable au sens de la distribution *a posteriori* (estimateur du MAP). Dans le cas ou cette distribution *a posteriori* est une distribution de Gibbs, il est en pratique impossible d'obtenir le MAP de manière analytique.

De la même manière, si la fonction de coût utilisée considère le nombre de pixels mal classés, l'approche bayésienne revient à chercher l'image segmentée dans laquelle les pixels sont mal classés avec une probabilité minimale (estimateur du MPM), et cette image ne peut pas s'obtenir directement.

Dans les deux cas, la minimisation du risque conditionnel (critère du MAP ou du MPM) nécessite l'utilisation d'algorithmes d'optimisation. L'approche la plus courante consiste à utiliser des méthodes de Monte Carlo associées à des procédures de recuit simulé (**?**). Notons que l'estimateur du MAP peut aussi être obtenu par des algorithmes déterministes. En effet trouver l'estimateur du MAP revient à maximiser l'énergie de la distribution de Gibbs *a posteriori*.

Le premier problème qui se pose dans le cas d'utilisation des méthodes de Monte Carlo est celui de la simulation d'images distribuées suivant la loi *a posteriori*. Le principe de toutes les méthodes existantes consiste à chercher une chaîne de Markov à état sur E, l'ensemble de toutes les images segmentées, irréductible et apériodique, d'état stationnaire limite unique $P(\mathbf{c}|\mathbf{x})$. En d'autre terme, si l'on note $\{\mathbf{C}^n : n = 1, 2, \cdots\}$ cette chaîne de Markov sur E, on a :

$$\lim_{n\to\infty} P(\mathbf{C}^n = \mathbf{c}|\mathbf{C}^0 = \mathbf{c}^0,\ \mathbf{x}) = P(\mathbf{c}|\mathbf{x})$$

Algorithme de Métropolis

Métropolis a proposé un algorithme dont chaque itération prend la forme suivante :

1. choix initial d'une image segmentée $\mathbf{c}^0$,
2. à l'itération n :
   - à partir d'une image $\mathbf{c}^n$ on tire une nouvelle image $\mathbf{c}^{n+1}$ suivant une certaine probabilité de transition. Cette probabilité devant être symétrique : $Q(\mathbf{c}^n|\mathbf{c}^{n+1}) = Q(\mathbf{c}^{n+1}|\mathbf{c}^n)$.
     Une stratégie consiste par exemple à sélectionner aléatoirement un pixel i dans l'image $\mathbf{c}^n$, puis à faire le choix d'un nouvel état k pour ce pixel.

L'état est tiré au hasard parmi les K possibles suivant une loi uniforme. $\mathbf{c}^{n+1}$ la nouvelle image est identique à $\mathbf{c}^n$ sauf éventuellement au pixel i ;

- Calcul du rapport :
$$\frac{P(\mathbf{c}^{n+1}|\mathbf{x})}{P(\mathbf{c}^n|\mathbf{x})} = r$$
- Si la nouvelle image est plus probable que l'ancienne ($r \geq 1$) alors on effectue la transition $\mathbf{c}^n \to \mathbf{c}^{n+1}$. Si la nouvelle image est moins probable que l'ancienne ($r < 1$), alors on génère un nombre u suivant une distribution uniforme entre 0 et 1 et on effectue la transition $\mathbf{c}^n \to \mathbf{c}^{n+1}$ si $u \leq r$.
De manière plus concise, $\mathbf{c}^n$ est remplacé par $\mathbf{c}^{n+1}$ avec une probabilité $p = \min(1, r)$.

Notons que le calcul du rapport r ne fait pas intervenir la constante Z et l'on peut montrer (**?**) que :
$$r = \frac{P(\mathbf{c}_i^{n+1}|\mathbf{c}_{G_i}^{n+1}, \mathbf{x})}{P(\mathbf{c}_i^{n}|\mathbf{c}_{G_i}^{n}, \mathbf{x})}$$
avec G_i dénotant les pixels voisins de i.

En résumé, l'algorithme de Métropolis simule une chaîne de Markov à état sur E dont la matrice de transition P est définie par
$$P_{cc'} = \begin{cases} Q_{cc'} \cdot \frac{P(\mathbf{c}'|\mathbf{x})}{P(\mathbf{c}|\mathbf{x})} & \text{si } P(\mathbf{c}'|\mathbf{x}) < P(\mathbf{c}|\mathbf{x}) \\ Q_{cc'} & \text{si } P(\mathbf{c}'|\mathbf{x}) \geq P(\mathbf{c}|\mathbf{x}) \text{ et } \mathbf{c}' \neq \mathbf{c} \\ 1 - \sum_{c':\ c' \neq c} P_{cc'} & \text{si } \mathbf{c} = \mathbf{c}' \end{cases}$$
La matrice de transition $Q = \{Q_{cc'}\}$ est choisit symétrique. On peut montrer que P sera irréductible dés que Q le sera. Comme P est réversible :
$$P(\mathbf{c}|\mathbf{x})P_{cc'} = P(\mathbf{c}'|\mathbf{x})P_{c'c}$$
La loi limite de la chaine sera bien la loi que l'on désire simuler.

Dynamique d'échange des spins

La dynamique d'échange des spins(**?**) est une version de l'algorithme de Métropolis qui simule une chaine irréductible sur un sous ensemble de E. L'image initiale $\mathbf{c}^0$ détermine le sous ensemble de E dans lequel évoluera l'image. Seule l'ensemble des images ayant le même nombre de pixels dans le même état que $\mathbf{c}^0$ sont accessible par la chaine de Markov simulée par la dynamique d'échange de spins.

La spécificité de la dynamique des spins réside dans la construction de l'image $\mathbf{c}^{n+1}$ à partir de l'image $\mathbf{c}^n$: deux pixels i et j de $\mathbf{c}^n$ sont choisis aléatoirement et on échange la valeur de leur réalisation pour créer $\mathbf{c}^{n+1}$. On comprend aisément que ce type d'échange n'autorise pas l'obtention d'image où le nombre de pixel dans un certain état est différent de celui de l'image initiale.

La transition $\mathbf{c}^n \to \mathbf{c}^{n+1}$ est accepté suivant le principe de Métropolis.

Échantillonneur de Gibbs

La simulation d'image par échantillonneur de Gibbs a été proposé par **?**). Cette méthode définit un ordre de visite des pixels et itère de la manière suivante :

1. choix initial d'une image segmentée $\mathbf{c}^0$,
2. à l'itération n :
   - un pixel i est choisi suivant l'ordre de visite,
   - ${\mathbf{c}_i}^{n+1}$ est tiré au hasard suivant la loi $P(\mathbf{c}_i|\mathbf{x};\mathbf{c}^n_{G_i})$.

Cet échantillonneur de Gibbs produit une suite d'images $\mathbf{c}^0,\cdots,\mathbf{c}^n$. Quand n est grand, on peut considérer que $\mathbf{c}^n$ est une réalisation de $P(\mathbf{c}|\mathbf{x})$. De plus on a la propriété suivante :

$$\lim_{n\to\infty}\frac{1}{n}[f(\mathbf{c}^0)+\cdots+f(\mathbf{c}^n)]=I\!E[f(\mathbf{C})],$$

avec f une fonction mesurable quelconque et $\mathbf{C}$ une variable aléatoire de loi $P(\mathbf{c}|\mathbf{x})$.

Recuit simulé et estimateur MAP

Si les méthodes d'échantillonage précédentes permettent d'obtenir des simulations suivant la loi souhaitée, elles ne permettent pas de déterminer directement l'image segmentée qui minimise le critère du MAP ou du MPM.

Dans le cas du MAP, on peut modifier les algorithmes précédents en intégrant une procédure de recuit simulé qui fera tendre la suite des images $\mathbf{c}^n$ vers le MAP.

L'idée consiste à introduire un paramètre de température T dans la distribution $P(\mathbf{c}|\mathbf{x})$, qui s'écrit

$$P(\mathbf{c}|\mathbf{x})=\frac{\exp(\frac{1}{T}\cdot U_{\Phi,\beta}(\mathbf{c},\mathbf{x}))}{Z(T)}.$$

À chaque étape la température décroît vers zéro. La convergence de l'algorithme est démontrée si la température décroît assez lentement. Cette décroissance lente a le désavantage de demander un très grand nombre d'itérations avant d'obtenir un estimateur du MAP satisfaisant.

Algorithme ICM

Pour pallier cette lenteur, **?**) propose un algorithme déterministe qui correspond à l'échantilloneur de Gibbs en prenant une température nulle dès le départ. Chaque itération de cet algorithme, baptisé ICM (Iterative Conditional Mode) modifie la classe d'un pixel de la façon suivante :

$${\mathbf{c}_i}^{n+1}=arg\max_{\mathbf{c}_i}P(\mathbf{c}_i|\mathbf{x};\mathbf{c}^n_{G_i}).$$

L'algorithme ICM a l'avantage de converger en moins de 10 examens de toute l'image et de faire croître $P(\mathbf{c}^n|\mathbf{x})$ à chaque itération. Le principal inconvénient de l'algorithme est sa forte dépendance par rapport aux conditions initiales.

Estimateur MPM

Une autre approche consiste à considérer le critère du MPM. Si l'on dispose d'un certain nombre de réalisations (d'images), de la loi *a posteriori*, alors les probabilités marginales (**?**) peuvent être estimées . En chaque pixel i, la fréquence empirique, m_{ik}, de la classe k est mesurée et fournit un estimation de la probabilité $P(\mathbf{C}_i = k|\mathbf{x})$.

Ainsi on peut obtenir une segmentation de l'image, raisonnablement bonne au sens du critère MPM, en classant chaque pixel comme suit :

$$c_{ik} = \begin{cases} 1 \ si \ k = arg\max_\ell m_{i\ell}; \\ \\ 0 \ sinon. \end{cases}$$

Les réalisation de la loi *a posteriori* peuvent être obtenue en utilisant un échantillonneur de Gibbs, l'algorithme de Métropolis ou bien une dynamique d'échange de spins.

7.5.6 Estimation supervisée

Lorsque l'on dispose d'une image bruitée et de l'image segmentée correspondante, plusieurs solutions existent pour estimer les paramètres (Φ, β). En fait ce problème se décompose alors en deux sous problèmes de nature identiques :

- estimer les paramètres β de $P(\mathbf{C} = \mathbf{c})$;
- estimer les paramètres Φ de $P(\mathbf{x}|\mathbf{c})$;

Comme ces deux problèmes reviennent à estimer les paramètres d'un champ de Markov connaissant au moins une réalisation de celui ci, nous nous limiterons au premier problème.

Vraisemblance

Dans la majorité des cas, il est impossible de calculer la vraisemblance d'un paramètre β donné. En effet, la vraisemblance s'écrit

$$\ell(\beta; \mathbf{c}) = \frac{\exp -U^a(\mathbf{c}, \beta)}{Z(\beta)}$$

où $Z(\beta)$ est incalculable car c'est une somme sur toutes les images segmentées possibles.

Dans le cas auto-normal, la constante de partition est connue et calculable, et les estimateurs du maximum de vraisemblance sont exploitables. Si l'on suppose que $\boldsymbol{\mu} = 0$ la vraisemblance d'un tel modèle prend la forme :

$$(2\pi\sigma^2)^{\frac{-1}{2}N} det(\mathbf{B})^{\frac{1}{2}} exp\{\frac{-1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{B}(\mathbf{x}-\boldsymbol{\mu})\}.$$

L'estimateur du m.v. de σ^2 est donné par

$$\hat{\sigma}^2 = N^{-1}\mathbf{x}\hat{\mathbf{B}}\mathbf{x},$$

et l'estimateur de $\mathbf{B}$ est obtenu en minimisant :

$$-N^{-1}\ln|\mathbf{B}| + \ln(\mathbf{x}^t\mathbf{B}\mathbf{x})$$

ce qui n'est pas un critère aisement minimisable à cause du calcul du déterminant de $\mathbf{B}$.

Pseudo-vraisemblance

Pour contourner le problème posé par la constante de partition, **?**) propose de trouver un estimateur qui optimise un critère calculable, la pseudo vraisemblance :

$$\beta^* = arg\max_{\beta} \prod_{i \in code\ell} P_\beta(\mathbf{c}_i|\mathbf{c}_{G_i})$$

où $code\ell$ est un ensemble de pixels conditionnellement indépendants et $\mathbf{c}_{G_i}$ est le voisinage du pixel i. Cette méthode a l'inconvénient de n'utiliser qu'une partie des données. Une extension assez naturelle consiste à utiliser tous les pixels même s'ils ne sont pas indépendants. Soit :

$$\beta^* = arg\max_{\beta} \prod_{i=1}^{N} P_\beta(\mathbf{c}_i|\mathbf{c}_{G_i}).$$

D'après **?**), ce critère donnerait des résultats plus fiables. Notons que même si la pseudo vraisemblance est facilement calculable pour une valeur donnée de β, l'obtention d'un β^* nécessite souvent l'utilisation d'algorithmes d'optimisation numérique.

Gradient stochastique

?) suggère une idée originale pour trouver un estimateur du maximum de vraisemblance de β. Soit $\mathbf{c}_o$ l'image segmentée disponible. Une condition nécessaire d'optimalité est

$$\nabla_\beta P_\beta(\mathbf{c}_o) = 0,$$

pour $\beta = \hat{\beta}_{MV}$. Cette équation peut se mettre sous la forme :

$$U^{a\prime}(\mathbf{c}, \beta) = I\!E[U^{a\prime}(\mathbf{C})]$$

où $U^{a\prime}(\mathbf{c})$ est le gradient de l'énergie U^a par rapport au vecteur β. Une montée de gradient stochastique peut alors être mis en œuvre pour résoudre cette dernière équation :

1. choix initial du vecteur β^0,
2. à l'itération m :
   - exécution d'une étape d'un échantillonneur de Gibbs qui simule $P_{\beta^m}(\mathbf{c})$; une nouvelle image $\mathbf{c}^m$ est obtenue,
   - calcul de
   $$\beta^{m+1} = \beta^m + \frac{\lambda}{m+1}[U^{a\prime}(\mathbf{c}^{m+1}) - U^{a\prime}(\mathbf{c}_o)]$$
   où λ est une constante.

La convergence de cet algorithme est démontrée, mais il est bien évident que le maximum atteint n'est que local.

7.5.7 Estimation non supervisée

Dans un contexte markovien, l'estimation des paramètres du modèle nécessite des réalisations d'images segmentées issues de ce modèle, et la segmentation nécessite la connaissances des paramètres du modèle. Pour résoudre ce problème, de nombreux algorithmes de segmentation non supervisée basés sur les champs de Markov sont des algorithmes itératifs qui utilisent un principe similaire à celui de l'algorithme EM :

1. choix initial de (β^0, Φ^0),
2. à l'itération m :
   - simulation d'une ou plusieurs images segmentées en utilisant le modèle de paramètres (β^m, Φ^m),
   - estimation de $(\beta^{m+1}, \Phi^{m+1})$ en utilisant l'image observée et une ou plusieurs images segmentées obtenues au cours des itérations précédentes.

Ces algorithmes sont trop nombreux pour que nous les détaillons tous (**?**, **?**, **?**, **?**, **?**). Nous donnerons donc un seul exemple proche des algorithmes proposés dans ce chapitre et qui illustrera les comparaisons numériques.

?) propose un algorithme baptisé EM Gibbsien qui est destiné à trouver les paramètres d'un modèle Markovien qui maximisent la pseudo vraisemblance et donne une image segmentée sur le principe du MPM dans le cas où l'on considère que le bruit est spatialement non corrélé et que les observations sont indépendantes conditionnellement à la connaissance des classes :

$$P_\Phi(\mathbf{x}|\mathbf{c}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\mathbf{c}_i),$$

La pseudo vraisemblance s'écrit alors :

$$\mathcal{P}_\Theta(\mathbf{c}, \mathbf{x}) = P_\Phi(\mathbf{x}|\mathbf{c}) \cdot \prod_{i=1}^{N} P_\beta(\mathbf{c}_i|\mathbf{c}_{G_i})$$

où $\Theta = (\Phi, \beta)$. S'inspirant de l'algorithme EM, l'algorithme prend la forme suivante :

1. choix initial du vecteur θ^0,
2. à l'itération $(m+1)$:
   - **Etape E :**
     - simulation d'une nouvelle série d'images $\mathbf{c}^1, \mathbf{c}^2, \cdots, \mathbf{c}^{s_o}, \cdots, \mathbf{c}^{s_m}$ suivant la loi $P_{\Theta^m}(\mathbf{c}_i|\mathbf{x}; \mathbf{c}_j, j \neq i)$ (s_0 est le nombre d'itérations requis pour que la suite $\{\mathbf{c}^m\}$ soit en régime stationnaire),
     - estimation des $u_{ik} = P_{\Theta^m}(c(\mathbf{x}_i) = k|\mathbf{x})$ (probabilité que $\mathbf{x}_i$ appartienne à la classe k conditionnellement à l'image observée) :
     $$u_{ik} = \frac{1}{s_m - s_0} \sum_{s=s_0}^{s_m} I\!\!I_{\{c_{ik}=1\}};$$
   - **Etape M :** Calcul de
     - $\Phi^{m+1} = arg\max_\Phi I\!\!E[\log P_\Phi(\mathbf{x}|\mathbf{c})|\mathbf{x}, \Phi^m]$. Dans le cas où le bruit est gaussien :
     $$\boldsymbol{\mu}_k^{m+1} = \frac{\sum_{i=1}^n u_{ik}}{n_k}; \tag{7.13}$$
     $$\boldsymbol{\Sigma}_k^{m+1} = \sum_{k=1}^K \sum_{i=1}^n \frac{u_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k^{m+1})(\mathbf{x}_i - \boldsymbol{\mu}_k^{m+1})^t}{n_k}; \tag{7.14}$$
     où $n_k = \sum_{i=1}^n u_{ik}$.
     - Au lieu de calculer
     $$\beta^{m+1} = arg\max_\beta I\!\!E[\log \prod_{i=1}^N P_\beta(\mathbf{c}_i|\mathbf{c}_{G_i}))|\mathbf{x}, \Phi^{m+1}],$$
     Chalmond calcule directement les probabilités $P_\beta(c_{ik} = 1|\mathbf{c}_{G_i})$. D'après la distribution *a priori* $P_\beta(\mathbf{c})$ choisit par l'auteur, il constate que la probabilité $P_\beta(c_{ik} = 1|\mathbf{c}_{G_i})$ prend un nombre fini de valeurs qui dépendent de la classe k du site i ainsi que de la configuration du voisinage entourant ce site. En notant $P(k|j)$ la valeur de la probabilité d'avoir le site i appartenant à la classe k, conditionnellement au voisinage j, Chalmond calcule les
     $$\hat{P}(k|j) = arg\max_{P(k|j)} I\!\!E[log \prod_{i=1}^N P_\beta(\mathbf{c}_i|\mathbf{c}_{G_i}))|\mathbf{x}, \Phi^{m+1}].$$
     Notons que ces valeurs sont utilisées dans l'itération E suivante pour déterminer les $P_\beta(c_{ik} = 1|\mathbf{c}_{G_i})$ qui servent à l'échantillonneur de Gibbs car
     $$P(c_{ik} = 1|\mathbf{x}; \mathbf{c}_{G_i}) \propto P_\Phi(\mathbf{x}_i|\Phi) \cdot P_\beta(c_{ik} = 1|\mathbf{c}_{G_i}).$$