REVUE DE STATISTIQUE APPLIQUÉE

G. GOVAERT

Classification binaire et modèles

Revue de statistique appliquée, tome 38, n° 1 (1990), p. 67-81

<http://www.numdam.org/item?id=RSA_1990__38_1_67_0>

Rev. Statistique Appliquée, 1990, XXXVIII (1), 67-81

CLASSIFICATION BINAIRE ET MODÈLES

G. GOVAERT

Université de Metz, Ile du Saulcy 57045 Metz
INRIA-Lorraine, BP 239 54506 Vandœuvre les Nancy Cedex

RÉSUMÉ

Les liens existant entre les méthodes de classification automatique et les modèles de statistique inférentielle ont surtout été étudiés lorsque les données sont quantitatives. Le critère d'inertie interclasse est alors associé à un mélange gaussien.

Nous nous proposons ici de le faire lorsque les données sont binaires. Nous montrons comment l'identification d'un mélange de distributions de Bernoulli avec le même paramètre pour toutes les classes et toutes les variables correspond à un critère de classification binaire utilisant la distance L_1 et des noyaux binaires. Nous avons généralisé ce modèle en prenant des paramètres qui dépendent des variables mais qui sont toujours les mêmes pour tous les classes. Enfin, nous terminons par le cas le plus général : cette fois, les paramètres peuvent varier suivant les classes et les variables. On retrouve le modèle des classes latentes traité par Celeux.

Mots-clés : *Classification binaire, distance L_1, mélange de lois de probabilité, modèle des classes latentes.*

ABSTRACT

The relations between clustering methods and statistical models have been studied when the data are continuous. Then the inertia criterion is associated with gaussian mixture. Here, we propose to study this relations when the data are binary. We show how the identification of a mixture of Bernoulli distributions with the same parameter for all classes and all variables correspond to a clustering criterion which uses L_1 distance and binary kernels. We have generalized this model, at first by selecting parameters which can depend on variables and finally by selecting parameters which can not only depend on the variables but also on classes. In this case, we obtain the latent class model.

Keywords : *Binary clustering, L_1 distance mixture, latent class models.*

Introduction

De nombreuses méthodes de classification reposent essentiellement sur la définition d'une métrique et d'un critère associé sans faire référence explicitement à des modèles probabilistes. En réalité, comme Celeux le propose (1988), et en particulier dans le cas du modèle gaussien, il est souvent possible de montrer qu'il y a un modèle sous-jacent. Celui-ci permet alors de donner une interprétation du critère et de justifier son choix. C'est ce que nous nous proposons de faire dans cet article avec une méthode simple de classification de données binaires.

Dans le premier paragraphe, nous décrivons cette méthode de classification de données binaires. Dans le second paragraphe, nous rappelons comment l'approche "classification" peut être utilisée pour identifier un mélange de lois de probabilité (Scott et Symons 1971, Schroeder 1976, Celeux 1988). Enfin dans le dernier paragraphe, nous proposons un modèle de mélange pour des données binaires et nous montrons que l'approche classification pour identifier ce modèle correspond à l'algorithme et au critère que nous décrivons dans le premier paragraphe. En outre, en s'appuyant sur des variantes de ce modèle, nous proposons des extensions de l'algorithme de classification de données binaires qui utilisent des distances pondérées par des coefficients variant suivant les classes et les variables (distance adaptatative binaire).

1. L'algorithme MNDBIN

La méthode de classification des Nuées Dynamiques (Celeux et alt. 1989) repose essentiellement sur l'utilisation de la notion de noyau associé à chaque classe. La nature de ce noyau peut être très variée. Dans le cas le plus simple et si les variables sont quantitatives, le noyau est un élément de l'espace R^p contenant l'ensemble à classifier. Lorsque les variables ne sont pas quantitatives, généralement on se ramène au cas précédent au moyen d'un changement de codage. Les noyaux fournis par la méthode des Nuées Dynamiques habituellement utilisés dans ce cas ont alors une structure différente des données initiales.

Nous proposons ici d'utiliser la possibilité d'ajouter facilement des contraintes aux noyaux pour lui imposer d'avoir exactement la même structure que les données initiales : les données étant binaires, chaque individu de l'ensemble à classifier peut être considéré comme appartenant à $\{0,1\}^p$. Il est donc naturel de vouloir représenter chaque classe par un élément de cet ensemble. Il reste alors à choisir une métrique sur cet espace $\{0,1\}^p$. Nous utilisons comme distance entre deux individus le nombre de valeurs qui ne sont pas identiques pour les deux vecteurs binaires correspondants. L'algorithme est alors particulièrement simple et le critère minimisé qui en résulte facile à interpréter.

1.1. Le problème

Ω étant un ensemble de n individus mesurés sur p variables binaires, on cherche une partition de Ω en K classes "homogènes", la valeur K étant donnée a priori.

On peut sans restriction supposer que les valeurs binaires sont 0 et 1. L'ensemble à classifier appartient donc à $\{0,1\}^p$.

1.2. La méthode des Nuées Dynamiques

Rappelons rapidement le principe de ces méthodes : On suppose que Ω est inclus dans un ensemble E (par exemple R^p); on définit un ensemble L de noyaux, une "distance" D entre les éléments de E et les noyaux de L. Le critère W de classification est alors le suivant :

$$W(P,L) = \sum_{k=1}^{K} \sum_{x \epsilon P_k} D(x, a_k)$$

où $P = (P_1, ..., P_K)$ est une partition de Ω en K classes

et $L = (a_1, ..., a_k)$ avec $a_k \epsilon L$

L'algorithme construit itérativement une suite $(P^0, L^0), (P^1, L^1), ..., (P^n, L^n)$ de partitions et de noyaux en minimisant à chaque étape le critère. On obtient ainsi à la convergence une partition avec comme résumé pour chaque classe le noyau associé.

Pour utiliser ce type d'algorithme dans le cas des données binaires, on peut soit considérer que l'ensemble des données appartient à R^p muni de la distance euclidienne et appliquer la méthode des Nuées Dynamiques en prenant comme noyau des éléments de R^p (méthode des centres mobiles), soit se placer simplement dans l'ensemble Ω muni d'une distance quelconque et utiliser la méthode des Nuées Dynamiques sur tableau de distances (ce qui revient à imposer aux noyaux d'appartenir à l'ensemble Ω). Les deux situations présentent des inconvénients. Dans le premier cas, le résumé de chaque classe et le critère sont difficilement interprétables par rapport aux données initiales; on n'a pas tenu compte de la forme particulière des données. Dans le second cas, cette fois l'appartenance des noyaux aux données initiales peut sembler trop restrictive et d'autre part cela va conduire à une certaine perte d'efficacité de l'algorithme en place mémoire et en temps (construction et utilisation du tableau de distances sur Ω).

Nous allons nous situer entre ces deux approches. pour ceci, nous utilisons la possibilité d'imposer des contraintes aux noyaux.

1.3. L'algorithme

Nous allons respecter un principe d'homogénéité : les données à classifier et les noyaux doivent être de même nature.

Les éléments de Ω appartenant à $\{0,1\}^p$, nous allons donc imposer aux noyaux d'appartenir à cet espace. Il suffit alors de choisir une distance sur l'espace $\{0,1\}^p$. La distance d que nous avons retenue est le nombre de fois où les coordonnées correspondantes ne sont pas identiques. On peut l'exprimer à l'aide de la distance L_1 ou distance "city-block" sur l'espace $\{0,1\}^p$:

$$\forall x \text{ et } y \epsilon \{0,1\}^p \quad d(x,y) = \sum_{j=1}^{p} \left| x^j - y^j \right|$$

où les x^j et y^j sont les coordonnées de x et y.

La distance $D(x, a_k)$ entre un élément x et un noyau a_k qui appartiennent tous les deux à $\{0,1\}^P$ est alors prise égale à $d(x, a_k)$.

Le critère que l'on cherche à minimiser s'écrit donc :

$$W(P,L) = \sum_{k=1}^{K} \sum_{x \epsilon P_k} d(x, a_k)$$

où $$P = (P_1, ..., P_K) \text{ et } L = (a_1, ..., a_K)$$

L'algorithme se construit alors de manière habituelle :

- *construction des classes (fonction f)* : on range chaque élément de Ω dans la classe dont le noyau est le plus proche.
- *construction des noyaux (fonction g)* : on associe à chaque classe P_k le vecteur binaire a_k minimisant

$$\sum_{x \epsilon P_k} d(x, a_k)$$

On a

$$\sum_{x \epsilon P_k} d(x, a_k) = \sum_{x \epsilon P_k} \sum_{j=1}^{p} \left| x^j - a_k^j \right| = \sum_{j=1}^{p} \sum_{x \epsilon P_k} \left| x^j - a_k^j \right|$$

a_k^j est donc l'élément majoritaire pour la variable j dans la classe P_k (on retrouve la notion de médiane). La construction du noyau est donc très simple.

1.4. Expression du critère à la convergence

A la convergence, le noyau étant fonction de la partition, on peut exprimer le critère uniquement par rapport à la partition. On obtient :

$$W'(P) = W(P, L) = W(P, g(P))$$

$$= \sum_{k=1}^{K} \sum_{j=1}^{p} \left| x^j - a_k^j \right| = \sum_{k=1}^{K} \sum_{j=1}^{p} A_j^k$$

où A_j^k est le nombre d'éléments minoritaires dans la classe P_k pour la variable j.

Ce critère représente le nombre de fois où la solution obtenue s'écarte de la situation "idéale".

1.5. Exemple

Soit un ensemble de 10 micro-ordinateurs identifiés par les lettres a à j et caractérisés par un ensemble de 10 propriétés identifiées par les nombres 1 à 10. On représente les données initiales (figure 1) sous forme d'un tableau binaire où un 1 indique que la propriété est vérifiée et un 0 qu'elle ne l'est pas.

Si on applique l'algorithme précédent en demandant 3 classes, on obtient comme partition de l'ensemble des micro-ordinateurs l'ensemble des classes {{a,d,h},{b,e,f,j},{c,g,i}}. On peut représenter cette partition sur les données initiales (figure 2) en réordonnant simplement les lignes de manière à respecter la partition. Les noyaux obtenus sont indiqués dans la figure 3 et le tableau des écarts A_j^k à la valeur idéale dans la figure 4. La valeur du critère est égale à 15, ce qui indique que sur 100 valeurs initiales du tableau, 15 ne sont pas égales à la valeur idéale représentée par le noyau correspondant.

	1	2	3	4	5	6	7	8	9	10
a	1	0	1	0	1	0	0	1	0	1
b	0	1	0	1	0	1	1	0	1	0
c	1	0	0	0	0	0	0	1	1	0
d	1	0	1	0	0	0	0	1	0	0
e	0	1	0	1	1	1	1	0	1	0
f	0	1	0	0	1	1	1	0	1	0
g	0	1	0	0	0	0	0	1	0	1
h	1	0	1	0	1	1	0	1	1	1
i	1	0	0	1	0	0	0	0	0	1
j	0	1	0	1	0	0	1	0	0	0

Fig.1
tableau initial

	1	2	3	4	5	6	7	8	9	10
a	1	0	1	0	1	0	0	1	0	1
b	1	0	1	0	0	0	0	1	0	0
h	1	0	1	0	1	1	0	1	1	1
b	0	1	0	1	0	1	1	0	1	0
e	0	1	0	1	1	1	1	0	1	0
f	0	1	0	0	1	1	1	0	1	0
j	0	1	0	1	0	0	1	0	0	0
c	1	0	0	0	0	0	0	1	1	0
g	0	1	0	0	0	0	0	1	0	1
i	1	0	0	1	0	0	0	0	0	1

Fig.2
tableau réordonné

A	1	0	1	0	1	0	0	1	0	1
B	0	1	0	1	0	1	1	0	1	0
C	1	0	0	0	0	0	0	1	0	1

Fig.3
les noyaux

A	0	0	0	0	1	1	0	0	1	1
B	0	0	0	1	2	1	0	0	1	0
C	1	1	0	1	0	0	0	1	1	1

Fig.4
tableau des écarts

1.6. Avantages de la méthode

Les données initiales sont résumées par K vecteurs binaires très facilement interprétables. La qualité du résultat, qui est fournie par la valeur du critère à la convergence, est simple à comprendre puisqu'il représente, nous l'avons vu, le nombre de différences entre les vecteurs binaires de départ et les vecteurs binaires caractérisant leur classe. Le tableau des valeurs A_j^k décrites précédemment permet de faire une analyse plus fine des résultats. Il représente en effet la décomposition du critère suivant les classes et les variables. Enfin, on peut montrer que tous ces résultats sont **indépendants du codage** binaire retenu, c'est-à-dire des valeurs numériques choisies pour coder les deux modalités binaires; en particulier on obtient exactement les mêmes résultats si on permute les valeurs 0 et 1.

1.7. Inconvénients de la méthode

On retrouve les inconvénients de toutes les méthodes de type Nuées Dynamiques, à savoir le problème du choix des éléments de départ et du nombre de classes.

2. L'approche classification de la décomposition de mélange

On reprend ici la présentation de Celeux (1988).

2.1. Identification d'un mélange

Le tableau de données de départ de dimension (n,p) est considéré comme un échantillon Ω de taille n d'une variable aléatoire à valeurs dans R^p dont la loi de probabilité admet la fonction de densité

$$f(x) = \sum_{k=1}^{K} p_k p(x/a_k) \qquad (1)$$

avec

$$\forall k = 1, K \quad p_k \epsilon]0,1[\quad \text{et} \quad \sum_{k=1}^{K} p_k = 1 \tag{2}$$

où $p(./a)$ appartient à une famille de fonctions de densité dépendant du paramètre a, élément de R^s où s est un entier supérieur ou égal à 1 et p_k est la probabilité qu'un point de l'échantillon suive la loi $p(./a_k)$. On appelera ces p_k les proportions du mélange.

Le problème posé est l'estimation du nombre k de composants et des paramètres inconnus $\{p_k, a_k / k = 1, K\}$ au vu de l'échantillon.

2.2. Approche classification

Dans l'approche classification (Scott et Symons 1971, Schroeder 1976), on remplace le problème initial d'estimation par le problème suivant :

Rechercher une partition $P = (P_1, ...P_k), K$ *étant supposé connu, telle que chaque classe* P_k *soit assimilable à un sous-échantillon qui suit une loi* $p(., a_k)$.

Il s'agit alors de maximiser le critère de **vraisemblance classifiante** :

$$W(P, a) = \sum_{k=1}^{K} \text{Log } L(P_k, a_k) \tag{3}$$

où a est le p-uplet $(a_1, ...a_k)$ et $L(P_k, a_k)$ est la vraisemblance du sous-échantillon P_k suivant la loi $p(./a_k)$: $L(P_k, a_k) = \prod_{x \epsilon P_k} p(x/a_k)$.

Pour maximiser le critère précédent, on peut utiliser des algorithmes de type Nuées Dynamiques qui construisent à partir d'une partition P^o en K classes une suite de partitions en appliquant les deux fonctions suivantes :

- une fonction de représentation g définie par $g(P) = g(P_1, ..., P_k) = (a_1, ..., a_k)$ où a_k est l'estimation du maximum de vraisemblance du paramètre de la densité associé au sous-échantillon P_k.
- une fonction d'affectation h définie par $h(a) = h(a_1, ..., a_k) = (P_1, ..., P_k)$ où $p_k = \{x \epsilon \Omega : p(x/a_k) \geq p(x/a_m)$ avec $k < m$ en cas d'égalité$\}$.

On peut alors montrer que sous certaines hypothèses, cet algorithme est convergent. On obtient à la convergence une partition P et une estimation des paramètres a_k. Les proportions p_k du mélange sont fournies par les fréquences des classes P_k.

Cette approche permet, par exemple, de donner une signification dans le cadre gaussien au critère d'inertie interclasse utilisé pour la classification d'individus décrits par des variables quantitatives.

3. Le modèle associé aux données binaires

3.1. La forme générale

On considère dans ce modèle que les données initiales forment un échantillon de taille n d'une variable aléatoire à valeurs dans $\{0,1\}^p$ dont la distribution de probabilité f est toujours définie par (1) et (2); mais ici $p(./a_k)$ est une distribution de probabilités sur $\{0,1\}^p$ appartenant à une famille paramétrée de distributions de probabilités.

En suivant l'approche "classification", rappelée dans le paragraphe précédent pour identifier le mélange, on se ramène à maximiser le critère de vraisemblance classifiante $W'(P,a)$ défini par (3).

On peut alors utiliser le même algorithme que dans le cas des données continues. A partir d'une partition P^o en K classes de l'échantillon, on applique successivement les deux fonctions g et h définies en 2.2 jusqu'à l'obtention d'une partition stable.

3.2. Choix de la famille de distribution

On suppose que, pour chaque composant du mélange, les p variables sont indépendantes et que chacune d'entre elles suit une des deux lois de Bernoulli suivantes :

$$\begin{cases} 1 \text{ avec la probabilité } 1-\varepsilon \text{ et } 0 \text{ avec la probabilité } \varepsilon \\ 1 \text{ avec la probabilité } \varepsilon \text{ et } 0 \text{ avec la probabilité } 1-\varepsilon \end{cases}$$

où $\varepsilon \epsilon]0, \frac{1}{2}[$, c'est-à-dire la loi de Bernoulli de paramètre $(1-\varepsilon)$ et la loi de Bernoulli de paramètre ε.

On peut alors écrire

$$p(x/a_k) = \prod_{j=1}^{p} \varepsilon^{|x^j - a_k^j|}.(1-\varepsilon)^{1-|x^j-a_k^j|}$$

où $a_k = (a_k^1, ..., a_k^p)$ et où les a_k^j indiquent quelle est la distribution retenue :

$$\begin{cases} a_k^j = 1 \text{ pour la première distribution} \\ a_k^j = 0 \text{ pour la seconde distribution} \end{cases}$$

Les paramètres à estimer sont donc les a_k^j et la valeur ε .

3.3. Lien avec la distance en valeur absolue sur $\{0,1\}^p$

Si on note d la distance en valeur absolue sur $\{0,1\}^p$, il est facile de montrer que l'on a

$$p(x/a_k) = \varepsilon^{d(x,a_k)}.(1-\varepsilon)^{(p-d(x,a_k))}$$

Le critère de vraisemblance classifiante s'écrit alors

$$\begin{aligned} W(P,a,\varepsilon) &= \sum_{k=1}^{K} \sum_{x \epsilon P_k} \text{Log} \left\{ \varepsilon^{d(x,a_k)}.(1-\varepsilon)^{(p-d(x,a_k))} \right\} \\ &= \sum_{k=1}^{K} \sum_{x \epsilon P_k} \{ d(x,a_k) . \text{Log}(\varepsilon) + (p - d(x,a_k)) . \text{Log}(1-\varepsilon) \} \\ &= \sum_{k=1}^{K} \sum_{x \epsilon P_k} \left(\text{Log} \frac{\varepsilon}{1-\varepsilon} \right) d(x,a_k) + n.p \text{ Log}(1-\varepsilon) \\ &= \left(\text{Log} \frac{\varepsilon}{1-\varepsilon} \right) \sum_{k=1}^{K} \sum_{x \epsilon P_k} d(x,a_k) + n.p \text{ Log}(1-\varepsilon) \end{aligned}$$

Cette expression montre que la recherche de ε et celle des a_k sont indépendantes. Pour un ε fixé, appartenant à $]0, \frac{1}{2}[$, $\text{Log}(\varepsilon/(1-\varepsilon))$ est négatif et maximiser le critère $W(P,a,\varepsilon)$ revient donc à minimiser le critère :

$$C(P,a) = \sum_{k=1}^{K} \sum_{x \epsilon P_k} d(x,a_k)$$

On retrouve ainsi le critère utilisé dans la méthode de classification de données binaires présentée dans le paragraphe 2.

On obtient donc une interprétation en terme probabiliste du critère de classification que nous avions proposé pour les données binaires :

- D'une part, on considère que les distributions des p variables sont indépendantes à l'intérieur de chaqueclasse, c'est-à-dire conditionnellement à l'appartenance à une classe. On retrouve ici l'hypothèse des classes latentes (Goodman 1974, Everitt, 1981).

- D'autre part, on suppose que les probabilités de toutes les distributions ne peuvent être que ε ou $1-\varepsilon$ où ε est un paramètre à déterminer. La valeur ε est en quelque sorte l'analogue de la matrice variance **commune** des classes construites par l'algorithme des centres mobiles.

  La valeur ε n'est pas intervenue dans cet algorithme qui minimise le critère

$C(P,a)$.S'il est nécessaire d'estimer ε, il suffit de maximiser le critère W' :

$$W'(P,a,\varepsilon) = \sum_{k=1}^{K} \sum_{x \in P_k} d(x,a_k) \left(\text{Log} \left(\frac{\varepsilon}{1-\varepsilon} \right) \right) + n.p \text{ Log}(1-\varepsilon).$$

Si l'on note e la valeur du critère $C(P,a)$ obtenu à la convergence de l'algorithme, on obtient :

$$W'(P,a,\varepsilon) = (n.p - e) \text{ Log}(1-\varepsilon) + e \text{ Log}(\varepsilon).$$

Il est facile de voir que la valeur ε maximisant cette quantité est $\frac{e}{n.p}$.

3.4. Généralisation à une distance L_1 adaptative : une seule distance

On se place exactement dans les mêmes conditions que dans le modèle précédent, mais en remplaçant la valeur réelle ε par un vecteur ε formé de p valeurs ε^j dépendant de chaque variable.

$$p\{x/(a_k,\varepsilon)\} = \prod_{j=1}^{p} \left\{ (\varepsilon^j)^{|x^j - a_k^j|} (1-\varepsilon^j)^{1-|x^j-a_k^j|} \right\} \text{ où } \epsilon^j \epsilon]0, \frac{1}{2}[.$$

Le critère de vraisemblance classifiante va alors s'écrire :

$$W(P,a,\varepsilon) = \sum_{k=1}^{K} \sum_{x \in P_k} \sum_{j=1}^{p} \left\{ \left(\text{Log} \frac{\varepsilon^j}{1-\epsilon^j} \right) \left| x^j - a_k^j \right| + \text{ Log} (1-\varepsilon^j) \right\}$$

$$= \sum_{k=1}^{K} \sum_{x \in P_k} \left\{ - \sum_{j=1}^{p} \left(\text{Log} \frac{1-\varepsilon^j}{\varepsilon^j} \right) \left| x^j - a_k^j \right| \right\} + n. \sum_{j=1}^{p} \text{Log} (1-\varepsilon^j)$$

Le critère se met alors sous la forme $- \sum_{k=1}^{K} \sum_{x \in P_k} d_\varepsilon (x,a_k) + A$ où d_ε est une distance de type L_1 pondérée par les quantités $\left(\text{Log} \frac{1-\varepsilon^j}{\varepsilon^j} \right)$ qui dépendent du vecteur ε et A est la quantité $n. \sum_{j=1}^{p} \text{Log} (1-\varepsilon^j)$.

Remarquons que l'on obtient bien une distance L_1 pondérée car les quantités $\left(\text{Log} \frac{1-\varepsilon^j}{\varepsilon^j} \right)$ sont nécessairement positives.

• *fonction d'affectation (recherche des classes)*

Le second terme A est constant lors de cette étape. Puisqu'on cherche à maximiser le critère W, on affectera les points aux "centres" les plus proches au sens d'une distance L_1 pondérée par les valeurs $\text{Log}\frac{1-\varepsilon^j}{\varepsilon^j}$.

• *fonction de représentation (recherche des a_k^j et des ε^j)*

Quelles que soient les valeurs ε^j obtenues, il est facile de montrer que les a_k^j sont nécessairement les valeurs majoritaires de chaque classe pour chaque variable. Il ne reste plus qu'à déterminer les ε^j. Il faut donc maximiser :

$$W(P,a,\varepsilon) = \sum_{k=1}^{K} \sum_{x \epsilon P_k} \left\{ -\sum_{j=1}^{p} \left(\text{Log}\frac{1-\varepsilon^j}{\varepsilon^j} \right) \left| x^j - a_k^j \right| \right\} \quad + \quad n. \sum_{j=1}^{p} \text{Log}\left(1-\varepsilon^j\right)$$

avec a_k^j valeur majoritaire.

On peut écrire :

$$W(P,a,\varepsilon) = -\sum_{j=1}^{p} \left(\text{Log}\frac{1-\varepsilon^j}{\varepsilon^j} \right) e_j \quad + \quad n. \sum_{j=1}^{p} \text{Log}\left(1-\varepsilon^j\right)$$

où e_j est le nombre de fois où la valeur majoritaire n'a pas été prise dans une classe pour la variable j.

$$W(P,a,\varepsilon) = \sum_{j=1}^{p} \left\{ (n-e_j) \left(\text{Log}\left(1-\varepsilon^j\right) \right) + e_j \text{Log}\left(\varepsilon^j\right) \right\} = \sum_{j=1}^{p} \phi\left(\varepsilon^j\right)$$

Il faut donc maximiser la fonction ϕ.

Pour ceci, on peut facilement vérifier que la valeur $\varepsilon^j = \frac{e_j}{n}$ qui annule la dérivée de la fonction ϕ

$$\phi'\left(\varepsilon^j\right) = \frac{-\left(n-e^j\right)}{1-\varepsilon^j} + \frac{e^j}{\varepsilon^j} = \frac{e_j - n.\varepsilon^j}{\left(1-\varepsilon^j\right).\varepsilon^j},$$

correspond à un maximum qui appartient bien à l'intervalle $]0, \frac{1}{2}[$ sauf dans le cas très particulier où $e_j = 0$.

Cette approche est similaire à la méthode des distances adaptatives avec une distance unique (Govaert 1975 et Diday et Govaert 1977).

On peut aller plus loin et généraliser encore en prenant une distance différente pour chaque classe.

3.5. Distance L_1 adaptative : cas général

On reprend le même modèle, mais cette fois avec des valeurs ε qui peuvent dépendre à la fois des classes et des variables.

$$p\{x/(a_k, \varepsilon_k)\} = \prod_{j=1}^{p} \left(\varepsilon_k^j\right)^{|x^j - a_k^j|} \cdot \left(1 - \varepsilon_k^j\right)^{1-|x^j - a_k^j|}$$

où $\varepsilon_k^j \epsilon]0, \frac{1}{2}[$.

Le critère de vraisemblance classifiante va alors s'écrire :

$$W(P, a, \varepsilon) = \sum_{k=1}^{K} \sum_{x \epsilon P_k} \left\{ -\sum_{j=1}^{p} \text{Log} \frac{1 - \varepsilon_k^j}{\varepsilon_k^j} \left| x^j - a_k^j \right| + \sum_{j=1}^{p} \text{Log}\left(1 - \varepsilon_k^j\right) \right\}$$

De manière analogue au paragraphe précédent, le critère se met encore sous la forme :

$$\sum_{k=1}^{K} \sum_{x \epsilon P_k} \{-d_{\varepsilon_k}(x, a_k) + A_k\}$$

où d_{ε_k} est une distance de type L_1 pondérée par les quantités $\text{Log} \frac{1 - \varepsilon_k^j}{\varepsilon_k^j}$ qui dépendent du vecteur ε_k des $\varepsilon_k^j (1 \leq j \leq p)$ et A_k est la quantité $\sum_{j=1}^{p} \text{Log}\left(1 - \varepsilon_k^j\right)$.

Les distances de type L_1 utilisées sont, comme dans le cas précédent, pondérées par des coefficients qui peuvent évoluer au cours des itérations de l'algorithme, mais cette fois, en plus, les distances sont différentes suivant les classes.

• *fonction d'affectation (recherche des classes)*

Cette fois le second terme dépend de k et aura donc une influence. On affectera x à la classe k qui minimise

$$d_{\varepsilon_k}(x, a_k) - \sum_{j=1}^{p} \text{Log}\left(1 - \varepsilon_k^j\right)$$

Remarque : sachant que $\sum_{j=1}^{p} \text{Log}\left(1 - \varepsilon_k^j\right) = \text{Log} \prod_{j=1}^{p} \left(1 - \varepsilon_k^j\right)$, si on impose aux coefficients de la distance la contrainte $\prod_{j=1}^{p} \left(1 - \varepsilon_k^j\right) =$ constante, on obtient cette fois une affectation que ne dépend que de la distance et nous nous retrouvons exactement dans la situation habituelle des Nuées Dynamiques. Ceci est à rapprocher de la comparaison entre la méthode des distances adaptatives dans le cas euclidien et la méthode de reconnaissance des mélanges de lois normales (Schroeder 1976). Dans le premier cas, une contrainte (le déterminant de la matrice définissant la métrique devait être constant) permettait d'utiliser l'algorithme habituel alors que

dans le second cas, où aucune contrainte n'était imposée, un terme additif venant en complément d'une métrique apparaissait dans le critère comme ici.

• *fonction de représentation (recherche des a_k^j et des ε_k^j)*

Comme dans les situations précédentes, quels que soient les coefficients ε_k^j, les meilleurs a_k^j sont les médianes par classe et par variable. Il reste à trouver les ε_k^j maximisant, pour chaque classe P_k et chaque variable j, la quantité :

$$C = \sum_{x \epsilon P_k} \left\{ - \left(\text{Log} \frac{1 - \varepsilon_k^j}{\varepsilon_k^j} \right) \left| x^j - a_k^j \right| + \text{Log} \left(1 - \varepsilon_k^j \right) \right\}$$

$$C = - \left(\text{Log} \frac{1 - \varepsilon_k^j}{\varepsilon_k^j} \right) c_k^j + n_k . \text{Log} \left(1 - \varepsilon_k^j \right)$$

où e_k^j est le nombre de fois où la valeur majoritaire n'a pas été prise dans la classe k pour la variable j et n_k le cardinal de la classe P_k.

$$C = \left(n_k - e_k^j \right) \text{ Log} \left(1 - \varepsilon_k^j \right) + e_k^j \text{ Log} \left(\varepsilon_k^j \right)$$

Le maximum est atteint pour $\varepsilon_k^j = \frac{e_k^j}{n_k}$.

En fait, avec cette dernière généralisation, on retrouve le modèle des classes latentes dans le cas le plus général traité par Celeux (1988).

Rappelons que l'hypothèse du modèle des classes latentes (Goodman 1974, Everitt 1981) est la suivante : il existe une variable qualitative ("cachée") à K modalités, telle que conditionnellement à la connaissance de l'une de ses modalités, les p variables soient mutuellement indépendantes. Le modèle s'écrit alors :

$$f(x) = \sum_{k=1}^{K} p_k p(x/\alpha_k)$$

avec $$\forall k = 1, K \quad p_k \epsilon]0,1[\ , \ \sum_{k=1}^{K} p_k = 1$$

et $$p(x/\alpha_k) = \prod_{j=1}^{p} \left(\alpha_k^j \right)^{x^j} . \left(1 - \alpha_k^j \right)^{1-x^j} \quad \text{où} \quad \alpha_k^j \epsilon]0,1[$$

On retrouve bien notre modèle; en effet nous avions

$$f(x) = \sum_{k=1}^{K} p_k . p(x/(a_k, \varepsilon_k))$$

avec

$$p(x/(a_k,\varepsilon_k)) = \prod_{j=1}^{p} \varepsilon_k^{j|x^j-a_k^j|}.\left(1-\varepsilon_k^j\right)^{1-|x^j-a_k^j|}$$

$$\text{où } \forall(j,k)\varepsilon_k^j \epsilon [0,\frac{1}{2}[$$

Il suffit de poser

$$\begin{array}{lllll} - \text{ si } \alpha_k^j \epsilon [0,\frac{1}{2}[& \varepsilon_j^j = \alpha_k^j & \text{et} & a_k^j = 0 \\ - \text{ si } a_k^j \epsilon]\frac{1}{2},1] & \varepsilon_k^j = 1-a_k^j & \text{et} & a_k^j = 1. \end{array}$$

Remarque :

Lorsque les données sont mises sous la forme d'un tableau disjonctif complet, Celeux (1988) a établi que l'identification d'un modèle de K classes latentes était équivalent à la recherche de la partition en K classes maximisant le critère d'information mutuelle.

D'autre part, il est connu que ce critère est très proche du χ^2 de contingence, critère maximisé par l'algorithme MNDQAL, le premier maximisant l'information mutuelle, le second le χ^2 de contingence. Dans ce cas, nous pensons que la première approche présente certains avantages : on travaille directement sur le tableau binaire sans avoir besoin de le dédoubler : les noyaux nécessairement binaires sont simples à interpréter; enfin on a une signification de l'algorithme en terme statistique, puisqu'il s'agit de l'identification d'un modèle de mélange de distributions de Bernoulli sous l'approche "classification".

Conclusion

Dans ce travail, nous avons établi comment l'algorithme de classification MNDBIN était lié à un modèle précis de mélange de données binaires. Ce lien permet d'expliquer les bons résultats que nous avions obtenus à l'aide de cet algorithme sur des données simulées qui justement suivaient ce modèle.

Cette approche permet aussi de justifier a posteriori l'utilisation pour les données binaires, d'une part de la distance en valeur absolue, d'autre part de noyaux binaires.

Enfin l'extension de ce modèle permet de proposer de nouveaux algorithmes utilisant des distances adaptatives de type L_1 qui restent à développer et à expérimenter.

Bibliographie

CELEUX G., DIDAY E., GOVAERT G., LECHEVALLIER Y., RALAMBONDRAINY H. (1989), Classification automatique des données : Environnement statistique et informatique. *Dunod* 1989.

CELEUX G. (1988), "Classification et modèles". *R.S.A.* Vol 36, n° 4, pp 43-57.

DIDAY E., GOVAERT G. (1977), "Classification avec distances adaptatives". *RAIRO* , V-11, n° 4, pp. 329-349.

EVERITT B. (1981), "An introduction to latent variable models". *Chapman and Hall.*

GOODMAN L. (1974), "Exploratory latent structure models using both identifiable and unidentifiable models". *Biometrika* 61.

GOVAERT G. (1975), "Classification Adaptative". *Thèse de 3ème cycle, Paris 6.*

SCHROEDER A. (1976), "Analyse d'un mélange de distribution de probabilité de même type". *R.S.A.*, Vol. 24, n° 1.

SCOTT A., SYMONS M. (1971), "Clustering methods based on likelihood ratio criteria". *Biometrics* 27.