Classification d'un tableau de contingence et modèle probabiliste

Gérard Govaert*,Mohamed Nadif**

*Heudiasyc, UMR CNRS 6599, Université de Technologie de Compiègne,
BP 20529, 60205 Compiègne Cedex, France
gerard.govaert@utc.fr
**LITA, Université de Metz, Ile du Saulcy, 57045 Metz Cedex, France
mohamed.nadif@univ-metz.fr

Résumé. Ces dernières années, la classification croisée ou classification par blocs, c'est-à-dire la recherche simultanée d'une partition des lignes et d'une partition des colonnes d'un tableau de données, est devenue un outil très utilisé en fouille de données. Dans ce domaine, l'information se présente souvent sous forme de tableaux de contingence ou tableaux de co-occurrence croisant les modalités de deux variables qualitatives. Dans cet article, nous étudions le problème de la classification croisée de ce type de données en nous appuyant sur un modèle de mélange probabiliste. En utilisant l'approche vraisemblance classifiante, nous proposons un algorithme de classification croisée basé sur la maximisation alternée de la vraisemblance associée à deux mélanges multinomiaux classiques et nous montrons alors que sous certaines contraintes restrictives, on retrouve les critères du Chi2 et de l'information mutuelle. Des résultats sur des données simulées et des données réelles illustrent et confirment l'efficacité et l'intérêt de cette approche.

1 Introduction

La classification automatique, comme la plupart des méthodes d'analyse de données peut être considérée comme une méthode de réduction et de simplification des données. Dans le cas où les données mettent en jeu deux ensembles I et J, ce qui est le cas le plus fréquent, la classification automatique en ne faisant porter la structure recherchée que sur un seul des deux ensembles, agit de façon dissymétrique et privilégie un des deux ensembles, contrairement par exemple à l'analyse factorielle des correspondances qui obtient simultanément des résultats sur les deux ensembles ; il est alors intéressant de rechercher *simultanément* une partition des deux ensembles. Ce type d'approche a suscité récemment beaucoup d'intérêt dans divers domaines tels que celui des biopuces où l'objectif est de caractériser des groupes de gènes par des groupes de conditions expérimentales ou encore celui de l'analyse textuelle où l'objectif est de caractériser des classes de documents par des classes de mots. Notons que dans ce domaine, les données se présentent généralement sous forme d'un tableau de contingence où chaque cellule correspond au nombre d'occurrences d'un mot dans un document.

Par ailleurs, les modèles de mélange de lois de probabilité (McLachlan et Peel, 2000) qui supposent que l'échantillon est formé de sous-populations caractérisées chacune par une distribution de probabilité, sont des modèles très intéressants en classification permettant d'une part de donner un sens probabiliste à divers critères classiques et d'autre part de proposer de nouveaux algorithmes généralisant par exemple l'algorithme classique des k-*means*. Dans le cadre de la classification croisée, on a pu ainsi montrer que l'algorithme $Crobin$ (Govaert, 1983) adapté aux données binaires peut être vu comme une version classifiante de l'algorithme *block EM* (Govaert et Nadif, 2005) dans un cas particulièrement simple de mélange de lois de Bernoulli.

Dans ce papier, nous proposons d'étendre ce travail à la classification croisée d'un tableau de contingence. Dans la section 2, nous définirons le modèle de mélange croisé adapté à ces données. La section 3 sera consacrée à la présentation de l'algorithme *Cemcroki2* dont l'objectif est la maximisation de la vraisemblance classifiante associée au modèle précédent. Nous montrerons dans la section 4 les liens de cet algorithme avec les critères du Chi2 et de l'information mutuelle. Dans la section 5, des résultats sur des données simulées et des données réelles confirmeront l'efficacité de cet algorithme et l'intérêt de notre approche qui peut être considérée comme une approche complémentaire de l'analyse des correspondances qui s'appuie sur la même représentation des données.

Notations Dans tout ce texte, on notera $\mathbf{x} = (x_{ij})$ le tableau de contingence construit sur les deux ensembles I et J ayant respectivement r et s éléments, $n = \sum_{i,j} x_{ij}$ la somme des éléments du tableau et $x_{i.} = \sum_j x_{ij}$ et $x_{.j} = \sum_i x_{ij}$ ses marges. On utilisera aussi le tableau des fréquences relatives $f_{ij} = x_{ij}/n$, $f_{i.} = \sum_j f_{ij}$ et $f_{.j} = \sum_i f_{ij}$ ses marges et les profils en ligne $f_J^i = (f_{i1}/f_{i.}, \ldots, f_{ir}/f_{i.})$. Une partition en g classes de l'ensemble I sera notée $\mathbf{z} = (z_{11}, \ldots, z_{ik}, \ldots, z_{ng})$ où $z_{ik} = 1$ si i est dans la classe k et $z_{ik} = 0$ sinon. Nous adoptons les mêmes notations pour la partition $\mathbf{w}$ en m classes de l'ensemble J. Par ailleurs, pour simplifier la présentation, les sommes et les produits portant sur I, J, $\mathbf{z}$ ou $\mathbf{w}$ seront indicés respectivement par les lettres i, j et k et ℓ sans indiquer les bornes de variation qui seront donc implicites. Ainsi, la somme $\sum_{i,j,k,\ell}$ portera sur toutes les lignes i allant de 1 à r, les colonnes j allant de 1 à s, les classes en ligne k allant de 1 à g et les classes en colonne ℓ de 1 à m.

2 Modèle de mélange croisé

Pour aborder le problème de la classification croisée sous l'aspect modèle de mélange, nous avons proposé (Govaert et Nadif, 2003) un modèle dont la densité s'écrit sous la forme

$$\sum_{(\mathbf{z},\mathbf{w})\in\mathcal{Z}\times\mathcal{W}} \prod_i p_{\mathbf{z}_i} \prod_j q_{\mathbf{w}_j} \prod_{i,j} \varphi_{\mathbf{z}_i\mathbf{w}_j}(x_{ij}; \alpha),$$

où les densités $\varphi_{k\ell}$ appartiennent à la même famille de densités de probabilité de $\mathbb{R}$, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ et $\rho = (\rho_1, \ldots, \rho_m)$ sont les proportions des classes k et ℓ et α est un paramètre qui dépendra de la situation étudiée. $\mathcal{Z}$ et $\mathcal{W}$ représentent respectivement les ensembles de partitions de I en g classes et de J en m classes.

Pour adapter ce modèle aux tables de contingence, on suppose que chaque valeur observée x_{ij} dans un bloc $k\ell$ de la table est la réalisation d'une variable aléatoire suivant une loi de Poisson de paramètre $\alpha_i\beta_j\delta_{k\ell}$ où les deux premiers termes expriment les effets en ligne et en colonne et le dernier correspond à l'effet du bloc $k\ell$.

La recherche d'une partition s'appuyant sur ce modèle consiste à maximiser la vraisemblance classifiante associée à notre modèle. Pour assurer l'identifiabilité du modèle, nous avons ajouté les conditions

$$\sum_{\ell} \beta_\ell \delta_{k\ell} = 1 \qquad \text{et} \qquad \sum_{k} \alpha_k \delta_{k\ell} = 1$$

où $\alpha_k = \sum_{i,k} z_{ik}\alpha_i$ et $\beta_\ell = \sum_{j,\ell} w_{j\ell}\beta_j$. Le problème de classification alors posé est de trouver les partitions $\mathbf{z}$ et $\mathbf{w}$ et le paramètre du modèle maximisant le critère

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_{i,k} z_{ik} \log \pi_k + \sum_{j,\ell} w_{j\ell} \log \rho_\ell + \sum_{i,j,k,\ell} z_{ik} w_{j\ell} x_{ij} \log \delta_{k\ell}$$

où $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \delta_{11}, \ldots, \delta_{gm})$ avec $\sum_\ell x_{.\ell}\delta_{k\ell} = 1$ et $\sum_k x_{k.}\delta_{k\ell} = 1$.

3 Algorithme de classification croisée

Pour maximiser $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$, nous proposons de maximiser alternativement cette fonction en fixant $\mathbf{w}$ et $\boldsymbol{\rho}$ puis $\mathbf{z}$ et $\boldsymbol{\pi}$. En posant $u_{i\ell} = \sum_j w_{j\ell} x_{ij}$, $u_{.\ell} = \sum_i u_{i\ell}$ et $\gamma_{k\ell} = u_{.\ell}\delta_{k\ell}$, on peut montrer que $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ se décompose en deux termes $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = L_c(\mathbf{z}, \boldsymbol{\theta}/\mathbf{w}) + g(\mathbf{x}, \mathbf{w}, \boldsymbol{\rho})$ où le premier correspond à une log-vraisemblance conditionnelle associée à un mélange de distributions multinomiales appliquées sur les échantillons $\mathbf{u}_1, \ldots, \mathbf{u}_r$ et le second terme ne dépend pas de $\mathbf{z}$. On peut alors utiliser l'algorithme CEM classique (Celeux et Govaert, 1992) pour obtenir la partition $\mathbf{z}$. En faisant un travail analogue pour la recherche de la partition $\mathbf{w}$, on obtient finalement l'algorithme *Cemcroki2* suivant :

1. Choix d'une position initiale $(\mathbf{z}^{(0)}, \mathbf{w}^{(0)}, \boldsymbol{\theta}^{(0)})$;
2. Répéter le calcul de $(\mathbf{z}^{(c+1)}, \mathbf{w}^{(c+1)}, \boldsymbol{\theta}^{(c+1)})$ à partir de $(\mathbf{z}^{(c)}, \mathbf{w}^{(c)}, \boldsymbol{\theta}^{(c)})$ jusqu'à la convergence :
   (a) Calcul de $\mathbf{z}^{(c+1)}, \boldsymbol{\pi}^{(c+1)}, \boldsymbol{\delta}^{(c+\frac{1}{2})}$ en utilisant l'algorithme CEM sur les données $(\mathbf{u}_1, \ldots, \mathbf{u}_r)$ à partir de $\mathbf{z}^{(c)}, \boldsymbol{\pi}^{(c)}, \boldsymbol{\delta}^{(c)}$.
   (b) Calcul de $\mathbf{w}^{(c+1)}, \boldsymbol{\rho}^{(c+1)}, \boldsymbol{\delta}^{(c+1)}$ en utilisant l'algorithme CEM sur les données $(\mathbf{v}1, \ldots, \mathbf{v}^s)$ à partir de $\mathbf{w}^{(c)}, \boldsymbol{\rho}^{(c)}, \boldsymbol{\delta}^{(c+\frac{1}{2})}$.

Les expressions des estimations des paramètres du modèle associés à chaque bloc $k\ell$ sont données par

$$\pi_k = \frac{\#z_k}{r}, \qquad \rho^\ell = \frac{\#w_\ell}{s} \qquad \text{et} \qquad \delta_{k\ell} = \frac{x_{k\ell}}{x_{k.}x_{.\ell}} = n\frac{f_{k\ell}}{f_{k.}f_{.\ell}}$$

où $\#$ représente le cardinal d'un ensemble.

4 Liens avec le Chi2 et l'information mutuelle

Après l'étape de maximisation, le critère s'écrit

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_k \#z_k \log \pi_k + \sum_\ell \#w_\ell \log \rho_\ell + n \sum_{k,\ell} f_{k\ell} \log \frac{f_{k\ell}}{f_{k.} f_{.\ell}} + cste$$

où $\sum_{k,\ell} f_{k\ell} \log \frac{f_{k\ell}}{f_{k.} f_{.\ell}}$ est l'information mutuelle associée au couple de partitions $\mathbf{z}$ et $\mathbf{w}$. De plus, en utilisant l'approximation $2x \log x \approx x^2 - 1$, on obtient aussi

$$L_c(\mathbf{z}, \mathbf{w}, \theta) \approx \sum_k \#z_k \log \pi_k + \sum_\ell \#w_\ell \log \rho_\ell + \frac{n}{2} \chi^2(\mathbf{z}, \mathbf{w}) + cste.$$

On peut ainsi observer que, lorsque les proportions sont fixées, la maximisation de $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ est équivalente à la maximisation de l'information mutuelle $I(\mathbf{z}, \mathbf{w})$ et approximativement équivalente à la maximisation du critère $\chi^2(\mathbf{z}, \mathbf{w})$: la maximisation du critère du $\chi^2(\mathbf{z}, \mathbf{w})$ utilisé par exemple dans l'algorithme *Croki2* (Govaert, 1983) ou de l'information mutuelle utilisée par exemple par Dhillon et al. (2003) supposent donc implicitement que les données sont issues d'un mélange croisé de distributions de Poisson avec des proportions égales et que l'algorithme que nous proposons peut être considéré comme une généralisation de ces algorithmes.

5 Expérimentations numériques

5.1 Données simulées

Pour illustrer le comportement de notre algorithme *Cemcroki2* et le comparer à l'algorithme *Croki2*, nous avons étudié leurs performances sur des données simulées. Nous avons sélectionné 48 types de données provenant d'un mélange croisé de Poisson à 3 classes en ligne et 2 en colonne ; nous avons retenu deux situations : proportions égales ($p_1 = p_2 = p_3$ et $q_1 = q_2$) ou non ($p_1 = 0.70, p_2 = 0.20, p_3 = 0.10$ et $q_1 = q_2$) et nous avons fait varier le degré de mélange ($5\%, 11\%, 16\%, 20\%, 27\%, 34\%$) et la taille des données ($r \times s = 30 \times 20, 50 \times 20, 100 \times 20, 500 \times 20$).

Pour chacun de ces 48 types de données, nous avons généré 30 échantillons et pour chaque échantillon, nous avons lancé les algorithmes *Cemcroki2* et *Croki2* 30 fois à partir de situations initiales aléatoires et sélectionné la meilleure solution. Afin de résumer le comportement des 2 algorithmes, nous avons utilisé le taux d'erreur de classification entre les partitions simulées et les partitions obtenues. Pour chaque algorithme et pour quelques exemples de degré de mélange, nous avons reporté dans les figures 1 et 2 les moyennes des taux d'erreur obtenus avec les 30 échantillons. Ces premières expériences montrent que dans toutes les situations, et en particulier pour des tailles d'échantillon suffisamment grandes, l'algorithme *Cemcroki2* donne de très bons résultats. Pour l'algorithme *Croki2*, on obtient de bons résultats uniquement pour des proportions égales.

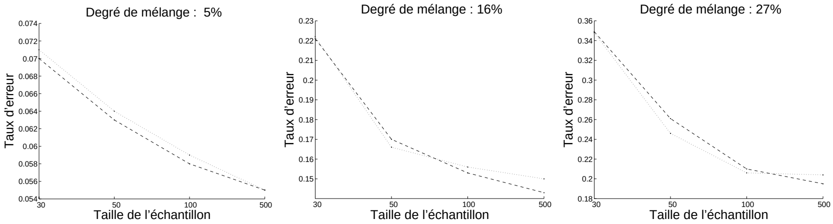

FIG. 1 – *Moyennes des taux d'erreur pour* Cemcroki2 *(ligne continue) et* Croki2 *(ligne pointillée) pour* $p_1 = p_2 = p_3$ *et* $q_1 = q_2$.

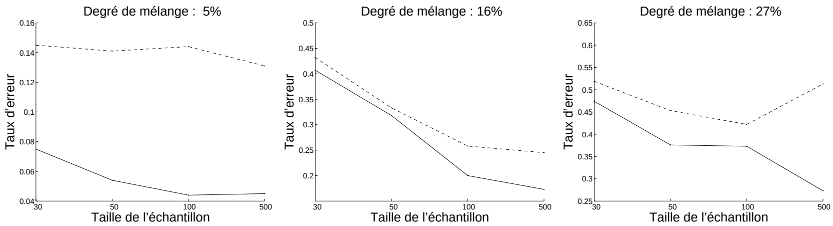

FIG. 2 – *Moyennes des taux d'erreur pour* Cemcroki2 *(ligne continue) et* Croki2 *(ligne pointillée) pour* $\mathbf{p} = (.70, .20, .10)$ *et* $q_1 = q_2$.

5.2 Données réelles

Pour illustrer l'algorithme *Cemcroki2* sur des donnés réelles, nous avons choisi les données SMART (ftp.cs.cornell.edu/pub/smart). Ces données sont définies à partir de 1033 résumés issus de la base Medline, de 1460 résumés issus de la base CISI et de 1400 résumés issus de la base CRANFIELD. En sélectionnant alors 2000 mots intéressants, Dhillon (2001) définit ainsi les données Classic3. Nous avons alors comparé les résultats obtenus par Dhillon (2001) et Dhillon et al. (2003) à l'aide de 2 algorithmes de classification croisée, que nous noterons *A2001* et *A2003*, avec ceux obtenus par notre algorithme *Cemcroki2*. La table 1 montre les matrices de confusion obtenues respectivement par *Cemcroki2, A2001* et *A2003*. Il apparaît clairement que *Cemcroki2* fournit les meilleurs résultats avec un nombre de documents mal classés de 49 contre 70 et 64 pour les algorithmes *A2001* et *A2003*.

	$Med.$	Cis.	Cra.	$Med.$	Cis.	Cra.	$Med.$	Cis.	Cra.
z_1	**1008**	23	2	**965**	0	0	**977**	22	34
z_2	2	**1453**	6	65	**1458**	0	1	**1444**	16
z_3	4	12	**1383**	3	2	**1390**	0	15	**1384**

TAB. 1 – Cemcroki2 *vs.* A2001 *et* A2003

6 Conclusion

En utilisant un modèle de mélange croisé de distributions de Poisson, nous avons proposé l'algorithme *Cemcroki2* et montré qu'il pouvait être vu comme une extension de *Croki2*. Ceci permet d'interpréter cet algorithme *Croki2* et d'en déduire par exemple que l'utilisation du χ^2 ou de l'information mutuelle supposent implicitement l'égalité des proportions des classes. Cette approche permet alors de prendre en considération de nouvelles situations comme celles où les proportions des classes sont très différentes. Les premières expériences sur des données simulées et réelles montrent que ce nouvel algorithme apparaît clairement meilleur que *Croki2* dans cette situation.

Références

Celeux, G. et G. Govaert (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis 14*(3), 315–332.

Dhillon, I. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Seventh ACM SIGKDD Conference*, San Francisco, California, USA, pp. 269–274.

Dhillon, I., S. Mallela, et D. Modha (2003). Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 89–98.

Govaert, G. (1983). *Classification croisée*. Thèse d'état, Université Paris 6, France.

Govaert, G. et M. Nadif (2003). Clustering with block mixture models. *Pattern Recognition 36*, 463–473.

Govaert, G. et M. Nadif (2005). An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 643–647.

McLachlan, G. J. et D. Peel (2000). *Finite Mixture Models*. New York : Wiley.

Summary

Most of methods of statistical analysis are concerned with understanding relationships among variables. With categorical variables, these relationships are usually studied from data that has been summarized by a contingency table, giving the frequencies of observations cross-classified by two variables. To classify the rows and the columns simultaneously of this contingency table, we can use *Croki2* which can be employed jointly with the correspondence analysis. In this paper, using a Poisson block mixture model, we have proposed the *Cemcroki2* algorithm which can be viewed as an extension of *Croki2*. In this setting, the probabilistic interpretation of *Croki2* constitutes an interesting support to consider various situations and avoids the development of ad hoc methods: for example, it allows one to take into account situations in which the proportions of clusters are different by applying *Cemcroki2* whereas the χ^2 and the mutual information criteria assume equal proportions implicitly. From our experiments, the new algorithm appears clearly better than *Croki2* in real situations when the proportions are not necessary equal.