Recherche de structures latentes dans des partitions de « textes » de 2 à K classes

Michèle Jardino

LIMSI – CNRS – BP 133 – 91403 Orsay – France
jardino@limsi.fr

Abstract

We have developed an original learning method in order to extract latent structures in raw texts. The induced structure is a data-driven tree which can be unbalanced. It has been obtained from successive partitions of the texts in classes, with an incremental number of classes ranging from 2 to k; each quasi-optimal partition has been performed with an adaptation of the k-means clustering. The paths of the texts in the successive partitions are the edges of an oriented graph whose nodes are the clusters. The study of the paths shows that some of the clusters remain identical in the successive partitions so that a tree can be extracted from the graph, by merging nodes and clipping edges. A corpus of 1100 touring information leaflets has been used to illustrate this method.

Résumé

Nous proposons une méthode automatique pour extraire une structure, si elle existe, dans des textes ou des ensembles de textes. Cette structure prend la forme d'un arbre de profondeurs et de ramifications variables qui sont déterminées par les données. Pour cela nous réalisons des partitions successives quasi-optimales des textes à l'aide d'un algorithme de classification non supervisée de type « centres de gravité mobiles ». Nous étudions ensuite les chemins effectués par les textes dans ces partitions successives. Ces chemins forment un treillis dans lequel on peut extraire par fusion et élagage un arbre si certaines classes restent stables dans les partitions successives. Nous illustrons cette recherche sur un corpus constitué d'un ensemble de 1100 fiches touristiques découpées en fragments de textes.

Mots-clés : structure latente de textes, arbre non équilibré, classification non supervisée, algorithme centres de gravité mobiles, théorie de l'information, entropie, recherche aléatoire.

1. Introduction

1.1. *Problème*

Étant donné un ensemble de « textes », par exemple des fragments de textes ou des pages Web, comment en faire une partition thématique non supervisée cohérente sans connaître a priori le nombre de classes et en faisant émerger une structure hiérarchique si elle existe ? Nous notons « textes » entre guillemets, car notre méthode est générique et peut être appliquée à tout type de données qui peuvent être représentées par des vecteurs (voir section 2.1).

1.2. *État de l'art*

L'abondance des textes électroniques disponibles aujourd'hui a donné un regain d'intérêt aux méthodes de classification non supervisées : ces méthodes automatiques sont adaptées à la gestion de grands flots de données. Deux types de partitionnement non supervisés sont généralement utilisés : la classification hiérarchique (ascendante ou descendante) qui permet de générer un arbre complet allant du regroupement de tous les éléments à classer dans une seule classe, à la racine de l'arbre, à la répartition de chaque élément dans sa propre classe, aux feuilles de l'arbre. L'autre méthode est le partitionnement direct en un nombre de classes spécifié à l'avance. Ces méthodes ont été développées dans différents logiciels commerciali-

sés, et sont disponibles gratuitement dans le logiciel libre, R (The R Project for Statistical Computing, http ://www.r-project.org/). Elles sont exposées dans de nombreux ouvrages, nous citerons Duda et Hart (1973), Kaufman et Rousseeuw (1990), Lebart *et al.* (1995). Dans ces méthodes, le choix du nombre de classes à choisir est un problème complexe, que ce soit pour la coupure de l'arbre généré par la classification hiérarchique ou pour le partitionnement direct. Il est généralement résolu soit par des connaissances a priori sur les données soit par une combinaison de méthodes de classification et d'analyse factorielle (Lebart *et al.*, 1995).

Dans tous les cas, cette coupure ne prend pas en compte la structuration réelle des données qui peuvent être éventuellement représentées par un arbre déséquilibré, avec des degrés de ramifications différents, comme par exemple pour les ontologies. Nous proposons ici une méthode pour trouver un tel arbre, si il existe, dans des données.

Proposition :

On recherche des classes qui restent stables d'une partition à une autre, c'est-à-dire des classes qui conservent les mêmes « textes ». Nous générons le treillis des relations observées entre classes de textes dans des partitions successives de 2 à K classes. L'analyse de ce treillis permet d'extraire, quand elle existe, une structure particulière sous forme d'arbre, sans connaissance a priori de cette structure, par fusion et élagage.

Nous avons étudié deux indices pour caractériser la stabilité des classes : le critère de partitionnement lui-même (Duda et Hart, 1973), et le nombre de chemins observés entre les différentes partitions.

Cette méthode peut paraître gourmande en temps de calcul car elle nécessite la recherche d'un ensemble de partitions successives. Elle bénéficie en fait d'un algorithme de classification rapide qui la rend aujourd'hui tout à fait accessible même pour traiter de grandes masses de données.

Nous décrirons d'abord comment nous représentons les textes et le corpus que nous avons utilisé pour tester cette méthode. Puis nous expliquerons notre algorithme de classification non supervisée. Enfin nous montrerons comment nous extrayons un arbre du treillis obtenu en illustrant par des graphes nos résultats. Nous mettrons ainsi en évidence les structures latentes dans ce corpus.

2. Représentations des textes

2.1. *Représentation vectorielle des textes dans l'espace des mots*

Pour représenter les textes, nous utilisons l'approche « sac de mots », largement utilisée en recherche d'information : un texte est décrit par l'ensemble des mots qu'il contient sans tenir compte de leur ordre. Ce qui est pris en compte c'est de savoir que les mots sont ensemble dans le texte, ce qui correspond à une notion de cooccurrence élargie : deux textes seront proches s'ils ont en commun un grand nombre de mots. Cette proximité peut se mesurer de manière binaire (présence ou absence) ou de manière plus précise en tenant compte de la fréquence des mots dans les textes. C'est le choix que nous avons fait : chaque texte est représenté par un vecteur dont les composantes sont les fréquences des mots normées par la longueur du texte, de sorte que la somme des composantes de chaque vecteur est égale à 1. Cette normalisation a été choisie car elle permet de représenter par un même vecteur les textes ayant une même distribution de mots comme par exemple un texte qui serait la concaténation d'un même texte répété plusieurs fois. Cette représentation est indépendante de la longueur du texte, elle correspond à la notion de profil utilisée en statistique (Lebart *et al.*, 1995).

En supposant qu'il y a T textes qui comportent au total V mots de vocabulaire différents, les textes sont indexés par i (i variant de 1 à T) et notés t_i, les mots par j (j variant de 1 à V) et notés m_j. Différentes notations du profil sont utilisées en statistique (Saporta, 1990 ; Lebart *et al.*, 1995), nous avons choisi de le noter f_i^j et d'utiliser une variante pour les fréquences marginales des textes, en remplaçant $f_{i.}$ par l_i car cette grandeur correspond à la longueur des textes et est plus facile à lire. Donc chaque texte t_i est représenté par un vecteur $\{ f_i^j \}$ dont les éléments f_i^j sont les fréquences relatives des mots dans le texte. On obtient ainsi une matrice de T lignes et V colonnes, chaque ligne correspond au profil d'un texte.

2.2. *Corpus d'étude*

Il s'agit de 1 100 fiches de descriptions de sites, de circuits ou de séjours touristiques. Afin de faciliter la recherche d'information dans ces fiches, par exemple pour répondre à la question « je cherche un séjour avec vue sur lagon », nous avons cherché à extraire automatiquement une structure de l'ensemble du corpus. Pour cela nous avons découpé chaque fiche en phrasettes puis regroupées celles-ci en classes thématiques selon la méthode décrite plus loin. Ce terme de phrasette au lieu de phrase a été choisi car la description des séjours dans les fiches est faite sans verbe et les segments de textes obtenus ne sont pas syntaxiquement des phrases. Les textes comprennent néanmoins des signes de ponctuation, ce qui nous a permis de les découper : chaque phrasette est un segment de texte compris entre deux points ou entre un début de description et un point. Finalement 4 700 phrasettes différentes ont été découpées dans ce corpus, à chaque phrasette est associé un vecteur. Le nombre total de mots de ce corpus est d'environ 120 000, la taille du vocabulaire est de 6 300 mots.

3. Partitions des textes

Nous présentons d'abord la méthode de partitionnement non supervisée en un nombre de classes prédéfini, ensuite nous montrons comment nous enchaînons les différentes partitions.

3.1. *Création d'une partition en k classes*

Nous disposons de T textes que nous voulons regrouper en classes homogènes bien séparées les unes des autres avec un algorithme qui permette de traiter un grand nombre de données en un temps raisonnable. Nous avons choisi la méthode de classification autour des centres mobiles (Lebart *et al.*, 1995) avec un critère de distance entropique et une recherche aléatoire de la meilleure classification (Jardino, 2000).

3.1.1. *Représentation des classes*

Chaque classe de textes est notée C_C (c varie de 1 à k) et est représentée par le vecteur associé au barycentre des textes de la classe. Ce vecteur correspond au profil de la classe de textes et sera noté f_c^j. On tient compte ainsi à la fois de la distribution des mots dans les textes (du profil des textes) et de la longueur de ces textes. L'expression du barycentre donne :

$$l_c \times f_c^j = \sum_{t_i \in C_c} l_i \times f_i^j$$

où l_c est la somme des longueurs des textes de la classe C_c et f_c^j la proportion du mot m_j dans la classe de textes C_c.

3.1.2. *Entropie, critère de classification*

Nous avons choisi comme critère de classification, l'entropie qui est une grandeur issue de la théorie de l'information (Cover et Thomas, 1991). Ce critère permet de déterminer la quantité d'information contenue dans les textes. Dans notre modèle de représentation « sac de mots », il est équivalent au nombre moyen de mots qui permettent de prédire un texte. L'entropie des

textes non classés est :

$$H(T) = -(1/f)\sum_{i=1}^{T}\sum_{j=1}^{V} f_i^j \times \log(f_i^j / l_i)$$

où f est le nombre total de mots du corpus.

Si chaque texte contient tous les mots de manière équiprobable, on a H(T) = - log(V) et on ne peut pas distinguer les textes les uns des autres. Si chaque texte est déterminé de manière unique par les mots qu'il contient on a H(T) = 0. Les valeurs d'entropie mesurées sur des corpus de textes réels sont comprises entre ces deux extrêmes.

L'entropie des textes regroupés en k classes représentées par leurs centres de gravité est :

$$H(k) = -(1/f)\sum_{c=1}^{k}\sum_{j=1}^{V} f_c^j \times \log(f_c^j / l_c)$$

On montre que l'entropie des textes regroupés H(k), quelque soit le regroupement, est plus grande que l'entropie des textes H(T) sauf dans le cas où le nombre k est égal au nombre de textes T et où chaque texte est seul dans sa classe, alors les deux entropies sont égales (Jardino, 2000).

Ce critère est intrinsèque aux centres de gravité des classes, et ne nécessite que de connaître les positions des centres de gravité et non pas leurs positions relatives, ce qui réduit les temps de calcul en comparaison de ceux obtenus avec un critère de distance ou de similarité.

3.1.3. *Algorithme de classification*

Pour une valeur de k donnée, la classification automatique cherche parmi les environ T^k configurations possibles (Jardino, 2000), celle qui minimise H(k).

Au lieu de rechercher d'emblée une classification optimale, nous avons choisi de rechercher au fur et à mesure une classification simplement meilleure que la précédente en cherchant aléatoirement les nouvelles configurations. En effet la recherche d'une solution optimale par la méthode du gradient conduit inéluctablement vers un optimum local, ce que nous voulons éviter. Cette recherche aléatoire a été inspirée par la méthode du recuit simulé (Jardino, 1993). L'objectif de cette méthode d'optimisation est d'éviter les minimums locaux en permettant d'accepter, dans un créneau fixé, des solutions qui vont en sens inverse du critère d'optimisation. En fait, comme l'ont remarqué différents utilisateurs, cette méthode bénéficie essentiellement de la recherche aléatoire des solutions et c'est finalement cet aspect que nous avons retenu. Ainsi nous obtenons des partitions qui sont quasi-indépendantes des conditions initiales.

En bref, l'algorithme de classification est le suivant :

– On choisit une partition initiale : à chaque texte est attribuée une classe puis les centres de gravité des classes sont calculés ainsi que l'entropie H(k) correspondante.

– De manière itérative, un texte est choisi au hasard et une nouvelle classe lui est attribuée également au hasard. Les centres de gravité de la classe initiale et de la nouvelle classe sont recalculés et la variation d'entropie associée s'en déduit. Si l'entropie décroît, le texte est affecté à la nouvelle classe, si elle croît, le texte reste dans la classe initiale. Le processus est réitéré jusqu'à ce qu'il n'y ait plus de variation d'entropie.

– En sortie nous disposons du classement final.

3.2. *Création de partitions de 2 à K classes*

Nous effectuons des partitions successives de 2 à K classes. La première partition est initialisée avec tous les textes regroupés dans une seule classe, les suivantes avec les résultats de classification obtenus dans la partition précédente : la partition 3, P3, est initialisée avec le classement obtenu lors de la partition P2. Les dénominations P2, P3, …, Pk, …, PK sont réservées aux partitions optimales en 2, 3,…, k, …, K classes.

Cette initialisation introduit un léger biais du fait que les partitions ne sont pas complètement optimales mais elle permet de mettre en évidence facilement les classes stables, car celles-ci conservent leur indice d'une partition à l'autre. Elle n'est pas indispensable, car comme nous le verrons ci-dessous, c'est la connaissance du chemin des textes dans les différentes partitions qui est importante.

3.3. *Partitions réalisées*

Des partitions de 2 à 10 classes ont été effectuées sur le corpus. Le tableau 1 suivant, rassemble des données et des résultats de partitionnement du corpus.

Nombre de textes, T	4 700 phrasettes
Taille du vocabulaire, V	6 300 mots
Nombre total de mots	120 000 mots
Entropie maximale	377 mots
Entropie minimale	25 mots
Entropie H(2)	295 mots
Temps de calcul	2 s
Entropie H(10)	162 mots
Temps de calcul	28 s

Tableau 1. Informations sur les partitions effectuées sur le corpus

Les entropies sont calculées en nombre moyen de mots qui prédisent les classes, cette valeur est calculée en prenant la valeur exponentielle de l'entropie, elle correspond à la notion de perplexité (Jelinek, 1988). L'entropie maximale est celle obtenue en rassemblant tous les textes dans une seule classe, on a donc un seul grand texte qui peut être prédit par 377 mots en moyenne. Cette dernière valeur est beaucoup plus petite que la taille du vocabulaire car elle tient compte de la redondance des mots. L'entropie minimale, 25 mots, correspond à la répartition des phrasettes dans 4 700 classes (une phrasette par classe).

Les temps de calcul sont courts, 2s pour une partition en 2 classes, 28s pour une partition en 10 classes.

4. Recherche d'un arbre dans le treillis

4.1. *Treillis*

Les partitions successives des textes de 2 à K classes fournissent un treillis à K niveaux. Il y a au maximum K ! branches qui connectent les classes dans les niveaux successifs. Un exemple est montré dans le tableau suivant, tableau 2, où sont recensés les chemins qui connectent le plus grand nombre de phrasettes dans les partitions de 2 à 10 classes. Chaque classe est représentée par un numéro allant de 1 à k pour chaque partition en k classes et chaque chemin est représenté par une suite de 9 chiffres, le premier chiffre de la suite varie entre 1 et 2, le deuxième entre 2 et 3 …

chemins	nombre de phrasettes ayant empruntées ces chemins
1 1 1 1 1 1 1 1 1	737
2 2 2 2 2 2 2 2 2	609
1 1 4 4 4 4 4 4 4	536
2 2 2 2 6 6 6 6 6	411
2 3 3 3 3 3 3 3 3	343
2 3 3 5 5 7 7 7 7	264
2 2 2 2 2 2 2 9 9	216
1 3 3 5 5 7 7 7 7	146
1 3 4 5 5 5 5 5 5	136
2 2 2 2 6 6 6 6 9	110

Tableau 2. Chemins les plus fréquentés par les phrasettes dans les partitions de 2 à 10 classes

Les phrasettes sont reliées par 295 chemins différents alors qu'il y a 10 ! possibilités (soit plus de 3 millions de chemins), ce qui montre qu'il y a des chemins privilégiés entre les différentes partitions. On peut remarquer que certaines classes n'apparaissent pas dans les 10 chemins les plus fréquentés : ce sont les classes 8 et 10. Aux dix premiers chemins sont associés 3 508 phrasettes, soit 75% du corpus.

Sur la figure 1 suivante, le treillis associé aux phrasettes est représenté sous forme de graphe. Cette figure montre comment se stabilisent les classes au fur et à mesure des partitions. On peut remarquer que la classe 1 est et reste une classe stable dès la partition en 4 classes. La classe 2 reste stable de la partition en 3 classes à la partition en 5 classes, elle se subdivise ensuite. La partition en 7 classes correspond à un niveau de stabilité pour toutes ses classes, car de cette partition il n'y a qu'une classe qui se partage pour créer la classe 8 dans la partition en 8 classes, les autres classes ne sont pas modifiées dans cette nouvelle partition.

4.2. *Extraction de l'arbre*

Visuellement, il est très facile de créer un arbre à partir des partitions. Il faut repérer des états stables et construire l'arbre à partir de ces états. Par exemple à partir du treillis précédent, si on s'arrête à la partition 8 on obtient l'arbre suivant, sur la figure 2. On part de la partition P8 de la figure 1 et on réalise des coupures dans l'arbre quand des branches s'isolent et des fusions quand une classe d'une partition est issue de plus d'une classe de la partition d'ordre immédiatement inférieure (voir détails dans la section suivante consacrée au critère de stabilité des classes). La fusion des classes 4, 5, 7 et 3 de P7 correspond à la fusion des classes 4, 5 et 3 des partitions P6 et P5 et à la fusion des classes 4 et 3 de P4, on leur associe donc un seul nœud dans la nouvelle arborescence qui sera étiqueté par les étiquettes des classes au niveau le plus proche de la racine soit ici : « 4,3 (P4) ». On fusionne dans cette branche les classes 1 et 3 de P3 dans une seule classe avec l'étiquette « 1,3 (P3) ». Les classes 6 et 8 de la partition P8 sont fusionnées dans la classe 6 des partitions P7 et P6, l'étiquette, établie selon la même règle que précédemment est « 6 (P6) ». Les classes 2 et 6 des partitions P7 et P6 fusionnent dans la classe 2 de la partition P5 que l'on retrouve identique dans P4 et P3, donc dans l'arbre avec l'étiquette « 2 (P3) ». La classe 1 de la partition 8 fusionne avec la classe 3 de la partition P3, elle porte l'étiquette « 1 (P4) » car c'est à partir de la partition P4 que cette classe s'isole définitivement.

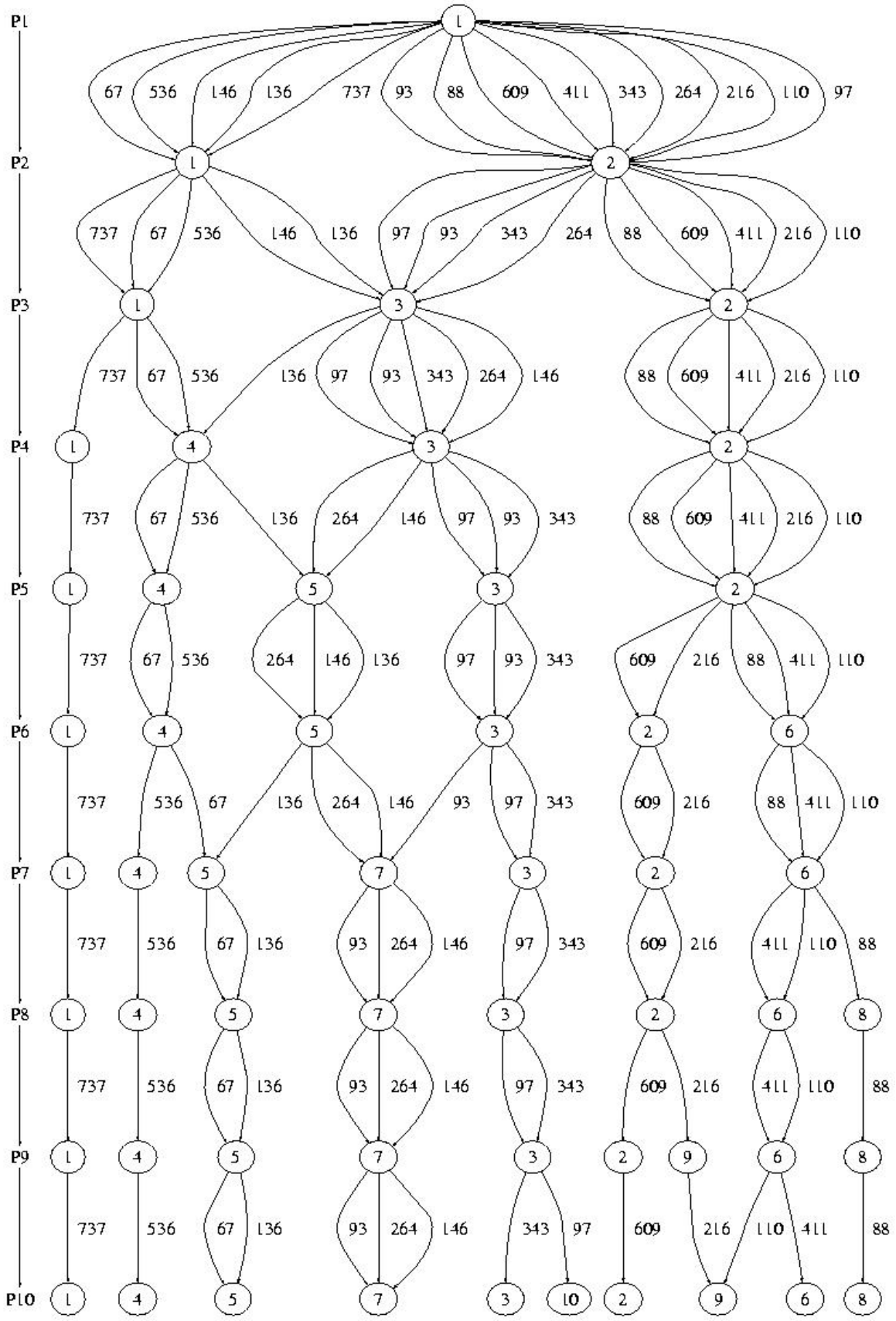

Figure 1. Treillis formé par les chemins des phrasettes dans les partitions de 2 à 10 classes

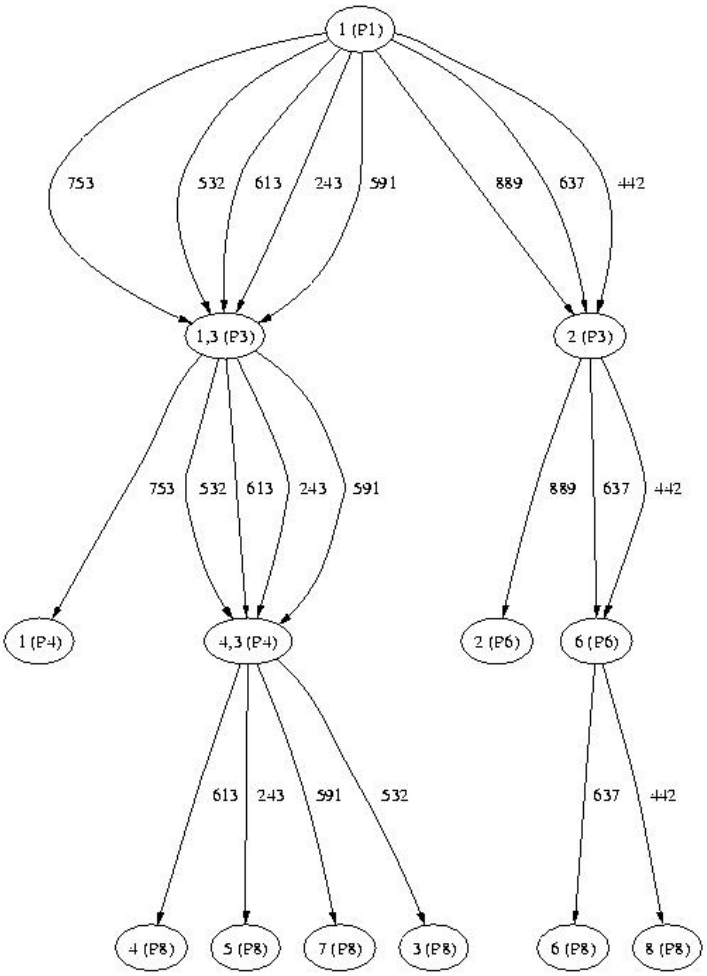

Figure 2.
Arbre extrait du treillis des chemins des phrasettes dans des partitions de 2 à 8 classes. Sur chaque branche figure le nombre de phrasettes qui transitent par les classes qui sont les nœuds de ce graphe

4.3. *Critères de stabilité des classes*

Des critères automatiques peuvent être utilisés pour trouver les classes stables dans les partitions successives. Nous en mettons deux en évidence, l'un fondé sur les chemins entre partitions, l'autre sur l'entropie des classes.

4.3.1. *Critère 1 : chemins entre deux partitions successives*

Une classe d'une partition en k classes sera dite stable si tous ses éléments (textes) sont issus d'une même classe de la partition en k-1 éléments, en d'autres termes, s'il n'existe qu'un chemin pour accéder à la partition k en venant de la partition k-1. Si toutes les classes de la partition k sont stables il y a k chemins entre les partitions k-1 et k, par contre si toutes les classes sont instables, il y a (k-1)*k chemins entre les partitions k-1 et k. Ainsi le nombre de chemins observés entre deux partitions successives peut être considéré comme un indicateur de stabilité des classes : pour une partition en k classes, il varie entre k (stabilité totale) et k*k-1 (instabilité totale), cet observable est un indice pour choisir une valeur de K.

Dans la phase exploratoire, il est nécessaire de choisir un nombre K assez grand pour observer s'il y a des zones de stabilité dans le nombre de chemins observés. A chaque texte on

associe la liste des classes auxquelles il appartient dans les partitions successives. On obtient ainsi tous les chemins observés que l'on compte ensuite. La courbe suivante, figure 3, montre l'évolution du nombre de chemins pour des partitions successives des phrasettes pour k variant de 1 à 15.

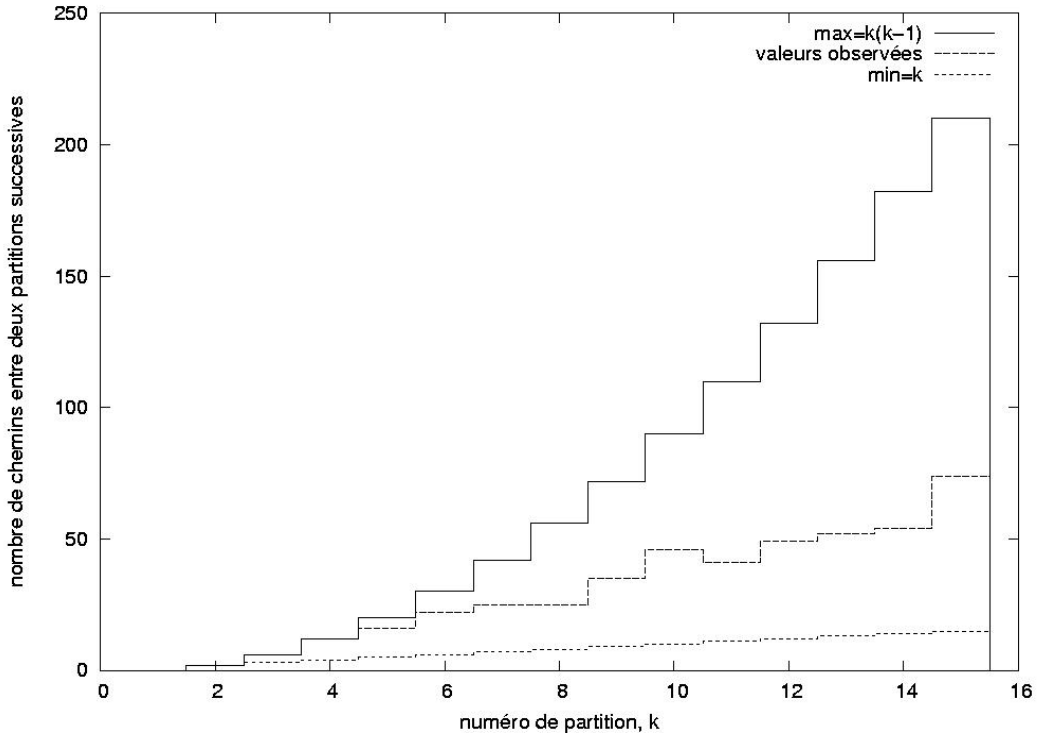

Figure 3. Nombres de chemins observés entre les partitions successives de phrasettes, comparés aux valeurs théoriques minimum et maximum

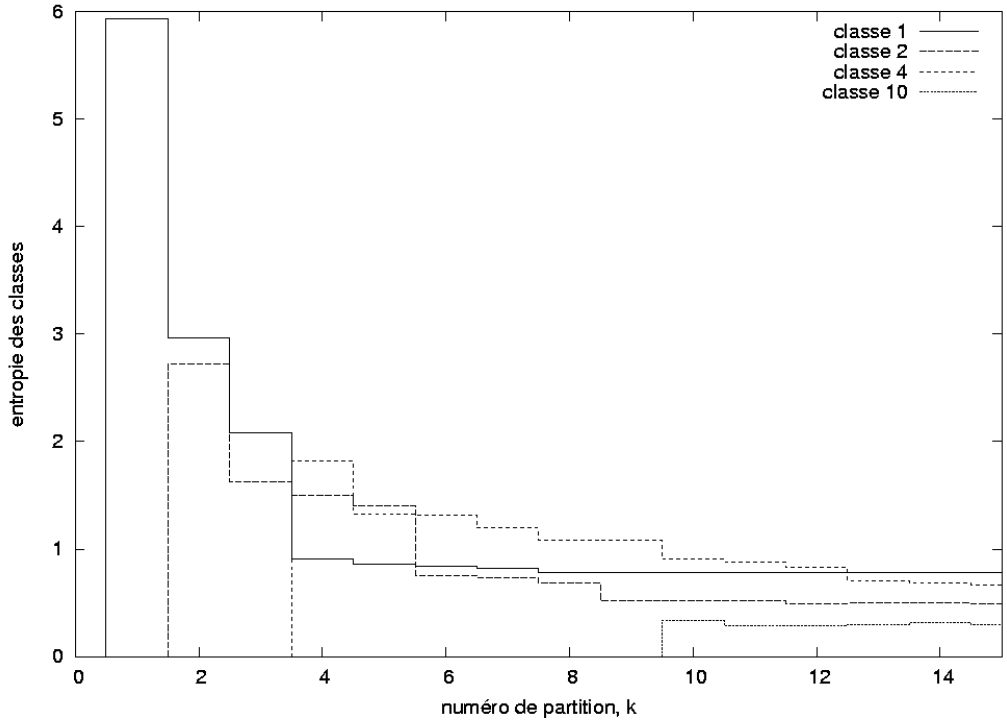

Figure 4. Variations d'entropie des classes 1, 2, 4 et 10 dans les partitions successives

On remarque qu'il y a stagnation du nombre de chemins entre les partitions P7 et P8 et une diminution de ce nombre entre P10 et P11, ce qui indique une stabilisation globale. La stagnation entre P7 et P8 correspond à la phase de stabilisation déjà observée dans le treillis de la figure 1.

4.3.2. *Critère 2 : entropie des classes*

On peut observer plus finement les transitions pour chaque classe, par exemple, comment évolue la classe 1 au travers des partitions, la classe 2 … Sur la figure 4 suivante nous montrons trois variations typiques de l'entropie dans les différentes partitions des phrasettes.

Pour la classe 1, l'entropie varie très rapidement en 3 étapes pour rester ensuite quasiment stable jusqu'à la partition en 15 classes.

Pour la classe 2, l'entropie varie fortement de la partition en 2 classes à la partition en 3 classes, elle varie peu entre les partitions de 3 à 5, puis de nouveau il y a une variation importante de P5 à P6, ensuite les variations sont faibles. On peut en déduire que la classe 2 est relativement stable dans les partitions 3, 4 et 5, que dans la partition 6, elle se divise pour redevenir stable dans les partitions 7 et 8. Elle se subdivise de nouveau à la partition 9 puis reste stable. La classe 2 semble être une classe homogène qui se subdivise.

L'entropie de la classe 4 chute fortement dans la partition P5, varie peu de P5 à P6 puis chute de nouveau avec P7. Les deux « chutes » correspondent à des divisions visibles sur le graphe de la figure 1 : division de la classe 4 (P4) en classes 4 (P5) et 5 (P6), de la classe 4 (P6) en classes 4 (P7) et 5 (P7).

Les variations d'entropie de la classe 10 montrent que des accroissements de l'entropie sont possibles au niveau local d'une classe.

On observe donc une variation de l'entropie fortement corrélée au parcours du graphe : de grandes variations correspondent à des divisions, des stagnations correspondent à des classes stables ou à des regroupements. Il est alors difficile avec ce critère de séparer ces deux derniers phénomènes. C'est d'ailleurs pour cela que la variation du critère d'optimisation qui est généralement utilisée (Duda et Hart, 1973) pour fixer le nombre de classes ne donne pas complètement satisfaction. Il faut plutôt le considérer comme un critère d'appoint par rapport au critère précédent.

5. Conclusion

Nous avons montré comment extraire une structure thématique latente dans des données : à partir de partitions successives de textes, de 2 à K classes, des classes stables dans ces partitions sont détectées et servent alors de point d'ancrage pour constituer un arbre éventuellement non équilibré avec des ramifications variées qui reflète la structure latente des données. On cherche d'abord un ordre de partitionnement (entre 2 et K) dans lequel les classes sont stables. A partir du treillis généré par les chemins des textes dans ces partitions successives, on génère un arbre par fusions et troncatures selon les branchements entre les partitions successives. Cette méthode originale peut être adaptée pour rechercher des classes stables à des niveaux de granularité plus fins, on peut utiliser d'autres fourchettes de variation, de K à K', et incrémenter ce nombre de classes par pas plus grand que 1. Elle doit également être complétée pour que la découverte de la structure soit entièrement automatisée à partir des critères de stabilité des classes décrits ci-dessus. Cette méthode non supervisée est utilisable sur tout type de données, par exemple, on peut envisager de créer une arborescence de mots à partir de profils de mots dans des textes ou de structurer des ensembles de pages Web.

Cette étude a été réalisée dans le cadre d'un projet BQR de l'Université Paris XI (Orsay, France) et a donné lieu à une première évaluation. Un fichier XML représentatif de la structure trouvée a été généré. A l'aide du formalisme de requêtes XPath, quelques questions ont été formulées pour tester l'apport de la hiérarchisation des informations touristiques. Dans tous les cas, la hiérarchisation améliore la précision de la recherche. Par exemple, pour la recherche de « séjour au bord d'un lagon », 19 fiches ont été trouvées dans la base de données XML pour 48 fiches retournées à partir des données non structurées. Les fiches supplémentaires correspondent à des « piscines à lagon » !

Références

Celeux G., Diday E., Govaert G., Lechevallier Y. et Ralambondrainy H. (1989). *Classification automatique des données.* Dunod.

Cover T. et Thomas J. (1991). *Elements of Information Theory*. Wiley & sons.

Duda R.O. et Hart P.E. (1973). *Pattern classification and Scene Analysis*. Wiley & sons.

Jardino M. et Adda G. (1993). Automatic word classification using simulated annealing. In *Proceedings of ICASSP'93*, Minneapolis, USA: 1191.

Jardino M. (2000). Unsupervised non-hierarchical entropy-based clustering. In Kiers H.A.L., Rasson J.-P., Groenen P.J.F. et Schader M. (Eds), *Data Analysis, Classification and Related Methods.* Springer : 29.

Jelinek F. (1988). *Statistical Methods for Speech recognition*. MIT Press.

Kaufman L. et Rousseeuw P.J. (1990). *Finding groups in data*.Wiley & sons.

Lebart L., Morineau A. et Piron M. (1995). *Statistique exploratoire multidimensionnelle.* Dunod.

Saporta G. (1990). *Probabilités, analyse des Données et Statistique*. Technip.