Classification automatique de structures arborescentes à l'aide du noyau de Fisher : Application aux documents XML

Ludovic DENOYER – Université Paris 6
Guillaume WISNIEWSKI – Université Paris 6
Patrick GALLINARI – Université Paris 6

Résumé : Le domaine de la Recherche d'Information Structurée (RIS) est un domaine qui émerge avec l'arrivée de données semi structurées comme les documents XML. Ce domaine, à travers l'initiative INEX, concerne principalement le développement de moteurs de recherche documentaire. Cependant, ,aujourd'hui, il est nécessaire de développer en parallèle des modèles pour le traitement de différentes problématiques dans les documents structurés comme la discrimination – qui vise à associer à chaque document une ou plusieurs étiquettes parmi un ensemble donné - ou la restructuration – qui cherche à projeter un ensemble de documents dans un schéma de médiation.

La difficulté principale rencontrée provient du fait qu'il existe très peu de mesures de similarité entre documents XML. Il est donc nécessaire aujourd'hui de s'intéresser à la conception de telles mesures.

Dans cet article, nous nous intéressons à la classification automatique de documents XML en fonction de leurs régularités structurelles. L'objet de cette problématique est de détecter automatiquement, à travers la structure des documents, l'ensemble des sources d'information dont ils sont issus. Cette problématique trouve son sens dans plusieurs applications ; elle peut permettre la visualisation par un utilisateur de l'organisation d'un corpus de documents hétérogènes comme le Web par exemple ; elle permet aussi de faciliter la recherche documentaire en sélectionnant la source qui, a priori, intéresse le plus un utilisateur.

L'article proposé ici a pour but de montrer comment un modèle génératif de documents structurés basé sur les réseaux bayésien[1] peut être utilisé, à travers le noyau de Fisher, comme un modèle permettant la mesure de la similarité entre deux documents XML. Cette similarité est évaluée à travers la tâche de classification automatique du corpus INEX.

Abstract: The widespread use of XML has urged the need to develop tools to efficiently store, access and organize XML corpus. The INEX initiative has resulted in major improvements in XML retrieval systems, but today, related tasks, like categorization or structure matching, should be investigated. We consider here the problem of clustering XML documents using their structure. In this paper, we propose a Belief networks-based stochastic model which is able to describe different kind of relation between structural elements. We show how these models can be used for the clustering task using the Fisher kernel method. We test them using the INEX corpus.

Introduction

Le développement du document électronique et du Web a vu émerger puis s'imposer des formats de données semi structurées, tels le XML et le XHTML. Ces nouveaux formats décrivent simultanément la structure logique des documents et le contenu de ceux-ci et permettent ainsi de représenter l'information sous une forme plus riche que le simple contenu. Celle-ci est adaptée à des besoins spécifiques qui permettent, par exemple, de faciliter l'accès à l'information ou d'optimiser le stockage et l'interrogation des documents. Avec l'augmentation rapide du nombre de documents semi structurés, il est nécessaire de concevoir de nouveaux modèles de Recherche d'Information (RI) capables de prendre en compte ce nouveau type de données, d'adapter les problématiques de la RI aux documents semi structurés et d'étudier les nouvelles problématiques que ces documents font émerger.

L'initiative INEX ([FUH 02]) étudie la problématique particulière de la recherche documentaire dans des grands corpus de documents XML. Les différents travaux menés dans le

[1] Le modèle génératif est présenté en détail dans l'article « Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML » de la conférence CORIA 2005

cadre cette initiative ont mis en évidence l'importance des problématiques connexes telles le traitement de données structurées hétérogènes. Une autre piste étudiée concerne la problématique de classification automatique qui, dans la RI traditionnelle, permet d'augmenter de manière significative la précision des moteurs de recherche. L'adaptation de cette problématique au traitement des données semi structurées, n'est pas triviale : doit-on considérer que deux documents sont proches lorsqu'ils possèdent des structures similaires, lorsque leur contenu est proche, ou seulement si leur structure et leur contenu sont proches ? La réponse à cette question dépend très fortement des applications considérées.

Les moteurs de recherche développés pour les documents XML sont dédiés au traitement de documents structurellement homogènes et font l'hypothèse que la structure des données est connue a priori par l'utilisateur. En particulier, les modèles développés dans le cadre d'INEX ne savent traiter que les documents dont la structure est régulière (étiquettes des nœuds identiques, même type d'informations...). Ainsi, il est important d'avoir à disposition des outils permettant de classer automatiquement de grands corpus afin de regrouper des documents de structure proche. C'est la problématique à laquelle nous nous intéressons dans cet article. Étant donné la taille des corpus de documents semi structurés, il est important que les modèles de classification automatique soient capables de traiter de grandes masses de données.

Dans cet article, nous présentons un modèle génératif de documents semi structurés, décrivant simultanément la structure et le contenu de ces documents. Ce modèle général peut être adapté à différentes tâches. Il a notamment été utilisé avec succès pour la discrimination ([DEN 04a]) et pour la restructuration automatique de documents XML ([DEN 04b]). Nous nous intéressons ici à la classification automatique de documents structurés et, plus particulièrement, au regroupement des documents ayant une structure proche. Le formalisme des réseaux bayésiens, sur lequel repose notre modèle, permet la prise en compte de différentes relations entre les éléments structurels d'un document. Nous voyons ensuite comment le modèle génératif proposé permet une représentation vectorielle des documents XML à l'aide du formalisme du noyau de Fisher et une réduction de la complexité de l'algorithme de classification automatique. Nous proposons de comparer différentes versions du modèle afin de mieux comprendre les relations pertinentes. Ces modèles sont testés sur la base de documents INEX.

1 État de l'art

La classification de documents XML est une problématique relativement nouvelle qui n'a reçu que peu d'attention, notamment parce qu'il n'existe pas, à ce jour, de corpus pour cette tâche.

[TER 02] propose d'utiliser la notion d'arbres fréquents pour réaliser un clustering de structures. L'algorithme *TreeFinder* permet de trouver, dans une collection d'arbres étiquetés non ordonnés, les arbres qui sont inclus dans au moins [epsilon] arbres de la collection. En utilisant différentes définitions de l'inclusion (inclusion stricte, inclusion ne conservant pas l'ordre des nœuds...), les auteurs arrivent à utiliser plusieurs modélisations de la structure d'un document. L'algorithme proposé réalise la classification du corpus selon la structure des documents, et associe à chacune des classes une structure représentative. [DOU 02] propose une méthode de classification utilisant à la fois le contenu et la structure des documents. C'est, à notre connaissance, la seule approche existante utilisant les deux types d'informations. Cette méthode propose de représenter les documents dans un espace vectoriel puis d'utiliser une méthode classique de classification vectorielle (les *k-means*). Les vecteurs représentants les documents sont composés de deux parties décrivant, respectivement le contenu et la structure à l'aide, respectivement, d'un codage tf/idf des mots et des étiquettes. Mais les résultats sont loin d'être concluants et les auteurs pensent qu'ils

n'arrivent pas à tirer pleinement parti des informations disponibles. [LEE 02] s'intéresse à une problématique proche de la notre : la classification, non pas de documents, mais de schémas et ne considère donc pas le même type de données que nous. D'autres travaux ont permis de développer des distances d'édition pour les arbres ([NIE 02]) qui pourraient être utilisées à l'aide d'algorithme classique pour la classification de structures. Toutefois, à cause de leur complexité, de telles méthodes ne sont pas adaptées au traitement de grande masse de données.

2 Modélisation de la structure d'un document

Nous avons développé un modèle génératif stochastique de documents semi structurés. Ce modèle repose sur le principe suivant : l'auteur va tout d'abord décrire *a priori* la structure (le plan) de son document puis « remplir » chacune de ces entités structurelles. Par exemple, pour la rédaction d'un article scientifique, l'auteur va décider qu'il doit y avoir un titre, un résumé, un certain nombre de sections composées de paragraphes, puis va rédiger le contenu de chacun des ces éléments. Selon la partie du document rédigée il ne va pas utiliser le même vocabulaire : la distribution du vocabulaire dépendra donc de l'entité structurelle.

Nous adoptons la représentation traditionnelle des documents semi structurés sous forme d'arbre ordonné. À chaque nœud du document correspond un nœud de l'arbre. La figure **Erreur ! Source du renvoi introuvable.** donne un exemple de représentation d'un document arborescent.

Étant donné un document d, nous noterons $|d|$ le nombre de ses nœuds. Chaque nœud n_i est composé d'une étiquette s_i et d'un contenu t_i et correspond à une entité structurelle du document (un paragraphe, un titre...). Soit Λ l'ensemble des étiquettes possibles (i.e. : $s_i \in \Lambda$). Le processus génératif décrit précédemment correspond à la modélisation probabiliste suivante (en utilisant un modèle de paramètres θ) : la probabilité de générer un document structuré est le produit de la probabilité a priori de la structure du document $P(s_1,...,s_{|d|}|\theta)$ et de la probabilité du contenu du document connaissant sa structure $P(t_1,...,t_{|d|}|s_1,...,s_{|d|},\theta)$. On a donc :

$$P(d \mid \theta) = P\left((s_1,t_1),...,(s_{|d|},t_{|d|}) \mid \theta\right) = P(s_1,...,s_{|d|} \mid \theta) \times P(t_1,...,t_{|d|} \mid s_1,...,s_{|d|},\theta) \quad (1)$$

Ce modèle a été dérivé pour les tâches spécifiques de discrimination de documents structurés ([DEN 04a]) et de restructuration de documents XML ([DEN 04b]). Il a montré sa capacité, d'une part, à traiter ces problématiques avec succès et, d'autre part, à traiter, aussi bien en apprentissage qu'en inférence, une quantité de données importante. Dans cet article, nous nous intéressons à sa capacité à classer automatiquement des structures sans considérer le contenu des documents. La probabilité d'un document ne dépend alors que de sa structure. On a donc :

$$P(d|\theta)=P(s_1,...,s_{|d|}|\theta) \quad (2)$$

Nous allons proposer différentes modélisations possibles des relations de dépendances entre les unités structurelles d'un document afin de déterminer les dépendances les plus intéressantes pour la classification automatique. Nous avons choisi de modéliser la structure par un réseau bayésien car ce formalisme permet de caractériser les dépendances conditionnelles entre variables aléatoires de manière flexible : à travers plusieurs topologies du réseau, nous allons pouvoir prendre en

compte différents types d'informations structurelles. Cependant, il est nécessaire de faire un compromis entre l'expressivité du modèle et sa complexité, pour pouvoir traiter une grande quantité de données.

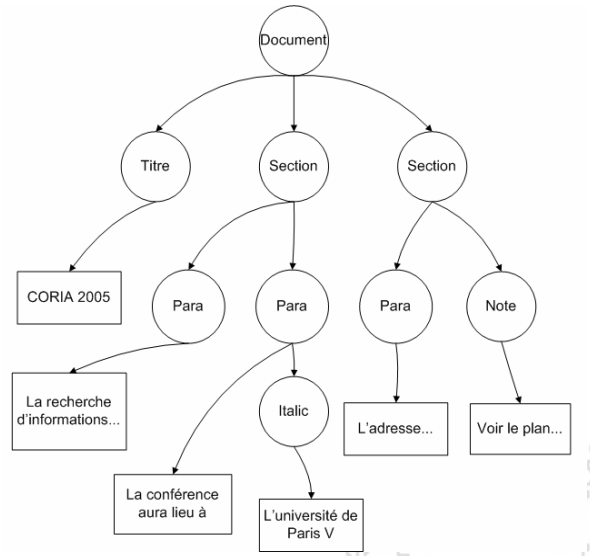


Figure 1: *Exemple de document semi structuré. Les informations de contenu apparaissent dans les rectangles ; les nœuds structurels sont représentés par des cercles dans lesquels apparaissent les étiquettes.*

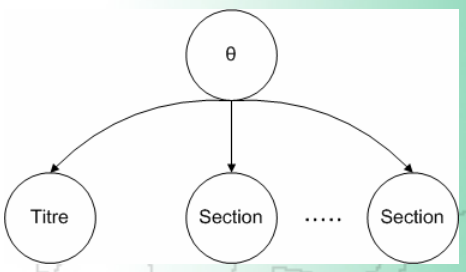


Figure 2: *Modélisation du document par un réseau bayésien de type Naïve Bayes*

3 Différents modèles de structure

3.1 Modèle général

Soit $(s_1,...,s_{|d|})$ l'ensemble des nœuds de structure d'un document d. On va considérer que la structure du document est modélisée par un réseau bayésien de N variables aléatoires $X_1,...,X_N$. Les arcs du réseau seront modélisés par la fonction $pa(X_i)$ qui renvoie l'ensemble des parents de la variables X_i dans le réseau. Nous allons distinguer deux types de variables :

- les variables $S_1,...,S_{|d|}$ qui correspondent à des nœuds du document modélisé. Ces variables seront à valeur dans Λ.
- les variables $Y_1,...,Y_{N-|d|}$ permettant de modéliser des dépendances supplémentaires entre les nœuds du documents. Ces variables ont pour but de modéliser des relations plus fines entres les élements de structure du document.

Ainsi, l'ensemble des variables s'écrit $(X_1,...,X_N)=(S_1,...,S_{|d|},Y_1,...,Y_{N-|d|})$. Nous allons proposer deux familles de modèles :

- la première famille (paragraphe 3.2) correspond à des réseaux « simples » pour lesquels toutes les variables aléatoires correspondent à des entités structurelles du document (i.e. : $N=|d|$). Cette famille permet la modélisation de dépendances directes entre les éléments d'un document.

- la seconde famille (paragraphe 3.3) permet de décrire, à l'aide des variables $Y_1,...,Y_{N-|d|}$ des dépendances supplémentaires.

3.2 Modèles de structure de type 1

On considère un ensemble de variables aléatoires $(S_1,...,S_{|d|})$ associées à chacune des parties d'un document structuré. La probabilité structurelle du document modélisé par un réseau bayésien d'arcs $pa(S_i)$ est obtenue par le calcul de la probabilité jointe du réseau :

$$P(d|\theta)=P(s_1,...,s_{|d|}|\theta)=P\left(S_1=s_1,...,S_{|d|}=s_{|d|}|\theta\right) = \prod_{i=1}^{|d|} P(S_i=s_i|pa(S_i),\theta) \quad (3)$$

La définition de la fonction $pa(S_i)$ permet de prendre en compte certaines relations structurelles.

3.2.1 Modèle de type « Naïve Bayes »

Le modèle de type *Naïve Bayes* considère l'indépendance des unités structurelles d'un document. Il correspond à un modèle de réseau où la fonction $pa(S_i)$ est la fonction vide. La figure **Erreur ! Source du renvoi introuvable.** donne le réseau construit pour un tel type de modèle. L'équation 3 se réécrit alors : $P(d|\theta)=\prod_{i=1}^{|d|} P(S_i=s_i|\theta)$. La probabilité $P(S_i=s_i|\theta)$ correspond alors à la probabilité qu'une partie s_i apparaisse dans le document — par exemple qu'il y ait un paragraphe dans un document. C'est un modèle simple de complexité faible, linéaire en fonction du nombre de nœuds du document.

3.2.2 Modèle parent

Le modèle *parent* vise à modéliser l'information d'inclusion entre les différentes entités structurelles d'un document. Il correspond à un réseau bayésien dans lequel $pa(S_i)=S_j$ si et seulement si le i-ème nœud du document *d* est le fils de son j-ème nœud. La figure 3 a donne le réseau correspondant à un tel type de modèle. La probabilité d'un document est alors :

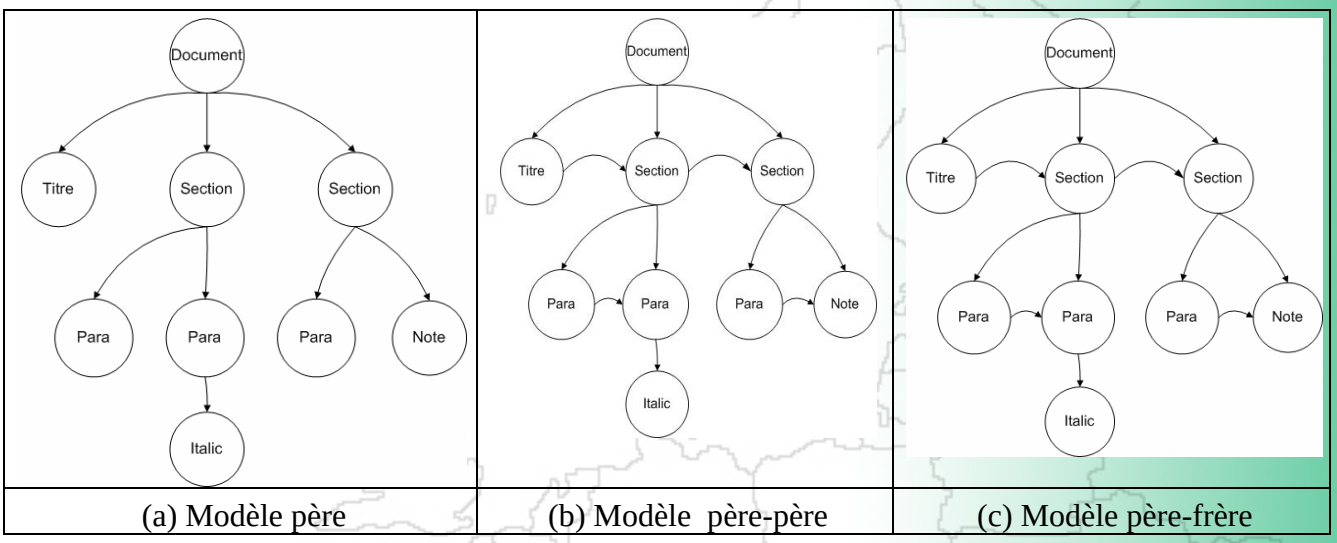
$$P(d|\theta)=\prod_{i=1}^{|d|} P\left(S_i=s_i|pa(S_i)|\theta\right)=\prod_{i=1}^{|d|} P\left(s_i|pere(s_i),\theta\right) \quad (4)$$

où $pere(s_i)$ est la fonction qui revoie l'étiquette du père du nœud *i* dans le document. La probabilité $P\left(s_i|pere(s_i),\theta\right)$ correspond à la probabilité qu'un nœud d'étiquette s_i soit le fils d'un

nœud possédant une étiquette *pere*(s_i) — par exemple, la probabilité d'avoir un paragraphe dans une section.

3.2.3 Autres modèles

Dans la même famille de modèle, nous proposons un modèle *grand-père* qui correspond à la modélisation de descendance d'ordre 2 (figure 3 b) et le modèle *père-frère* (figure 3 c) qui correspond à la modélisation de la relation d'inclusion et de la relation de séquentialité : un nœud *paragraphe* est dans une *section* et apparaît après une *introduction*. Nous ne détaillons pas ces modèles qui ressemblent beaucoup au modèle de type parent.



(a) Modèle père (b) Modèle père-père (c) Modèle père-frère

Figure 3 : *Les réseaux bayésiens décrivant les dépendances conditionnelles entre les différents nœuds du document (la dépendance avec les paramètres θ n'est pas représentée).*

3.3 Modèle grammaire

Nous proposons ici un modèle de structure plus complexe qui vise à modéliser la dépendance entre un nœud et l'ensemble de ses fils, contrairement aux modèles précédents qui établissent une dépendance entre chacun des nœuds et leur parent ou voisin. Nous verrons plus loin que ce modèle permet d'extraire, sous forme d'une DTD probabiliste, un représentant structurel d'un corpus.

Cette modélisation de la structure est inspirée des grammaires probabilistes d'arbre ([CAR 02]). La structure de l'arbre est décrite par une grammaire de type CFG, associant à chaque nœud la liste ordonnée de ses fils. Ainsi, la règle de dérivation *document*→*titre section section* indique qu'un nœud d'étiquette *document* aura trois enfants d'étiquette respective : *titre*, *section* et *section*. Nous considérons ici le processus génératif dans lequel l'auteur d'un document structuré découpe un document en plusieurs parties étiquetées puis, récursivement, redécoupe chacune de ces parties en sous-parties. Le réseau bayésien correspondant à l'équation précédente est représenté à la figure 4 Soit (n_i) l'ensemble ordonné des étiquettes des enfants d'un nœud du document. Nous associerons à chaque règle un nœud Y_i=(n_i) dans le réseau bayésien décrivant le document. Deux types de probabilités seront alors à considérer :

- celles du type $P(Y_i|X_i,\theta)$ qui décrivent la probabilité que l'auteur utilise la règle $X_i \rightarrow Y_i$ pour découper le nœud X_i
- celles du type $P(X_i|Y_i,\theta)$ qui décrivent la probabilité que l'on trouve un nœud X_i sous un nœud Y_i

Nous supposerons que pour, tout *i*, $P(X_i|Y_i,\theta)=1$: le choix de la règle de réécriture détermine de manière certaine les nœuds se trouvant sous un nœud *Y*. La probabilité structurelle d'un document calculée par un tel modèle stochastique s'exprime alors par :

$$P(s_1,...,s_{|d|} \mid q) = \prod_{t=1}^{|\delta|} P\left(enfant(n_t) \mid s_t, \theta\right)$$

3.5 Modèle génératif et similarité

Nous ne détaillons pas dans cet article les méthodes d'apprentissage de ces modèles (voir [DEN04c]. **La probabilité $P(d/\theta)$ correspond à une mesure de similarité entre le document d et l'ensemble des documents utilisés pour l'apprentissage des paramètres θ .** En ce sens, le modèle proposé, dans sa version générative permet la mesure de similarité entre documents et entre un document et un groupe de document. Dans notre problématique de classification automatique, nous avons utilisé un modèle de mélange afin d'attribuer un cluster à chaque document. Nous ne détaillons pas ici le modèle de mélange utilisé (voir [DEN04c]).

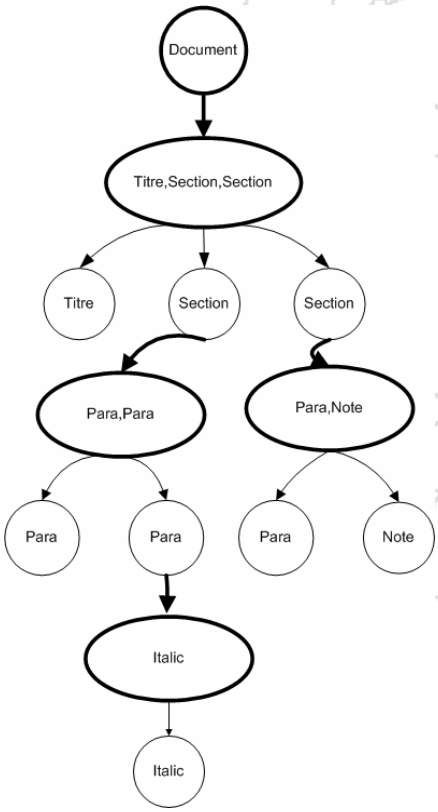


Figure 4: Le réseau bayésien décrit le modèle *grammaire*. Les flèches épaisses correspondent

$$U_d = \left(\nabla_\Theta \log P(s^d/\Theta^s), \nabla_\Theta \left(\sum_{i/s_i^d=l_1} \log P(t_i^d/s_i^d,\Theta_{ti}^t) \right), ..., \nabla_\Theta \left(\sum_{i/s_i^d=l_{|\Lambda|}} \log P(t_i^d/s_i^d,\Theta_{ti}^t) \right) \right)$$

$$\nabla_\Theta \log P(t^d/s^d,\Theta^t)$$

$l_{|\Lambda|}$

Figure 5 : Vecteur de Fisher. *Le vecteur obtenu est une composition d'un vecteur représentant l'information de structure (partie gauche) et d'un ensemble de vecteurs représentant les*

aux probabilités $P\left(enfants(n_i)|s_i,\theta\right)$.

contenus pour les différentes étiquettes possibles des nœuds.

3.6 Noyau de Fisher

La méthode du noyau de Fisher a été développée au départ pour la classification de séquences biologiques ([JAA98]). Son but est de transformer un modèle génératif en modèle discriminant afin d'accroître ses performances pour la tâche de classification. L'idée de ce modèle consiste à créer à l'aide d'un modèle génératif une fonction noyau qui pourra ensuite être utilisée dans différentes machines discriminantes comme les MVS par exemple.

Soit un modèle génératif $P(d/\Theta)$, Jaakkola propose de calculer le score de Fisher du document *d* :

$$U_d = \nabla_{\Theta} \log P(d/\Theta)$$

Où l'opérateur ∇_{Θ} représente le gradient par rapport à Θ. U_d est alors un vecteur dont la dimension est égale au cardinal de Θ. En ce sens, U_d est une représentation vectorielle du document d par rapport à un modèle génératif de paramètres Θ. Chaque composante du vecteur représente combien le paramètre du modèle génératif contribue à générer l'exemple donné.

A l'aide de ce score, nous pouvons alors définir une similarité entre deux exemples d_1 et d_2 à l'aide du noyau de Fisher suivant :

$$K(d_1,d_2) = U_{d_1}^T M^{-1} U_{d_2}$$

Où $M = E_X[U_X^T U_Y]$.

Dans cet article, nous proposons d'utliser la représentation vectorielle des documents obtenue à l'aide du noyau de Fisher et des modèles génératifs présentés précédemment dans un algorithme classique de classification automatique (algorithme des k-moyenne dans notre cas). Ainsi, **la similarité entre deux documents sera obtenue en calculant le produit scalaire entre les vecteurs normalisés représentant les documents. Nous ne détaillons pas ici l'algorithme des k-moyenne.**

3.6.1 Application du noyau de Fisher au modèle précédent

En utilisant la décomposition structure-contenu, le vecteur de Fisher s'écrit :

$$\begin{aligned} U_d &= \nabla_{\Theta}\left(\log P(s^d/\Theta^s) + \log P(t^d/s^d,\Theta^t)\right) \\ &= \nabla_{\Theta} \log P(s^d/\Theta^s) + \sum_{l\in\Lambda} \nabla_{\Theta}\left(\sum_{i/s_i^d=l} \log P(t_i^d/s_i^d,\Theta_u^t)\right) \end{aligned}$$

On en déduit donc que la représentation vectorielle d'un document structuré sera la somme des vecteurs de Fisher correspondant à chacune des composantes du modèle. Ces composantes seront :

- un vecteur représentant la structure du document d (i.e : $\nabla_{\Theta} \log P(s^d/\Theta^s)$)

- un vecteur représentant le contenu (i.e : $\nabla_\Theta \log P(t^d / s^d, \Theta^t)$) lui-même composé d'un sous-vecteur par étiquette possible des nœuds (i.e $\nabla_\Theta \left(\sum_{i / s_i^d = l} \log P(t_i^d / s_i^d, \Theta_{il}^t) \right)$)

La figure 5 représente de manière graphique le type de vecteur obtenu par le modèle général.

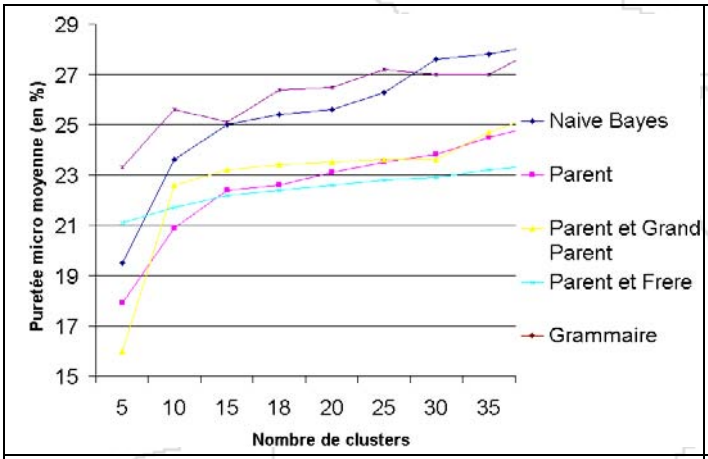
Finalement, la représentation de Fisher revient à coder dans différents morceaux d'un vecteur des informations issus des différentes entités structurelles d'un document. Elle est ici issue d'une dérivation naturelle de notre modèle génératif à l'aide de la méthode des noyaux de Fisher.

4 Classification automatique

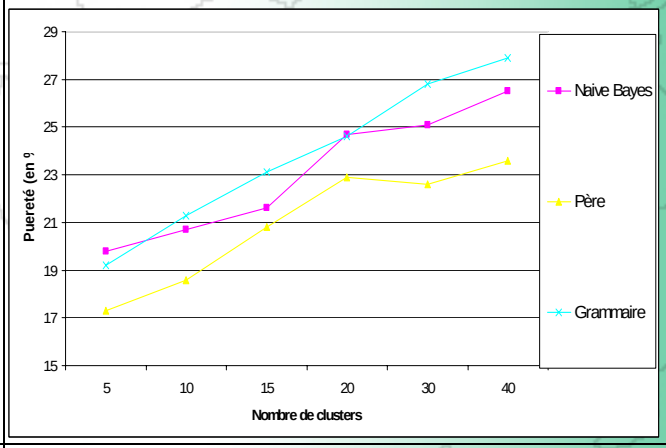
4.1 Expériences

De nombreuses manières d'évaluer un système de classification ont été proposées dans la littérature. [ZHA 01] présente une synthèse des différentes méthodes utilisées. Nous avons choisi d'utiliser comme mesure d'évaluation la mesure de *pureté* qui mesure la proportion de documents d'une classe ayant l'étiquette la plus fréquente. De manière générale, l'évaluation d'un système de classification automatique est un problème difficile et les mesures proposées ne permettent qu'une appréciation de la qualité de la classification.

Le corpus INEX a été rassemblé dans le cadre d'une campagne d'évaluation des moteurs de recherche XML et regroupe près de 12 000 articles scientifiques au format XML ce qui représente plus de 7 000 000 de nœuds. Ces documents proviennent de 18 journals. Nous allons regarder comment notre méthode de classification de structures permet de séparer les documents provenant des différents journaux.



(a) Modèle génératif (modèle de mélange)



(b) Modèle vectoriel
(modèle de Fisher et algorithme k-moyenne)

La figure précédente détaille les résultats obtenus sur le corpus INEX. On s'aperçoit que la prise en compte de relations relativement simples comme la relation d'inclusion (modèle *parent*) ou la relation de séquentialité (modèle *père-frère*) ne permettent pas d'obtenir, sur ce corpus, de

meilleurs résultats que le modèle simple *Naïve Bayes*. Ce résultat contre intuitif provient notamment du fait que le corpus utilisé n'est pas adapté à la classification automatique. Cependant, les expériences proposées sont, à notre connaissance, les premières effectuées sur INEX, le seul corpus réel de grande taille de documents XML existant à l'heure actuelle. Par contre, le modèle *grammaire* montre de meilleures performances, notamment lorsque le nombre de classes est faible. Il n'existe de pas de différence significative entre les performances du modèle vectoriel et celles du modèle génératif. Il est cependant important de noter que la complexité du modèle vectoriel est beaucoup plus faible que celle du modèle génératif car ce modèle ne nécessite pas l'estimation, à chaque itération des paramètres du modèle génératif, et nous privilégierons donc l'usage de la méthode vectorielle basée sur le noyau de Fisher.

5 Conclusion

Nous avons proposé un modèle génératif de documents structurés qui a trouvé précédemment son application dans les domaines de la discrimination et de la restructuration de documents arborescents. Dans cet article, nous proposons un formalisme pour le calcul de la probabilité de la structure des documents arborescents. Ce formalisme basé sur les réseaux bayésiens est flexible et permet de spécifier différentes dépendances entre les unités structurelles des documents. Le modèle *grammaire* qui modélise les dépendances entre un nœud et l'ensemble de ces fils se révèle être le plus performant en classification automatique sur le corpus INEX. Nous avons montré comment obtenir un système moins coûteux du point de vue du temps de calcul à l'aide du formalisme du noyau de Fisher. La tâche de classification de structure est une tâche émergente dans la communauté de la Recherche d'Information Structurée et ce travail propose un ensemble d'expériences sur la modélisation statistique et l'utilisation d'un tel système sur une base réelle et de grande taille : la base INEX.

6 References

[CAR 02] CARRASCO R. C.RICO-JUAN J. R., A similarity between probabilistic tree languages: application to XML document families, *Pattern Recognition*, , 2002.

[DEN 04a] DENOYER L.GALLINARI P., Bayesian Network Model for Semi-Structured Document Classification, *Information Processing and Management*, , 2004.

[DEN 04b] DENOYER L., WISNIEWSKI G.GALLINARI P., Document Structure Matching for heterogeneous corpora, *SIGIR 2004*, Workshop on IR and XML, Sheffield, 2004.

[DEN 04c] DENOYER L., Apprentissage statistique dans les bases de documents structurés. Thèse de doctorate *2004*, Université Paris 6.

[DOU 02] DOUCET A.AHONEN-MYKA H., Naïve clustering of a large XML document collection, *Proceedings of the First INEX Workshop.*, 2002, 81–87.

[FUH 02] FUHR N., GOVERT N., KAZAI G.LALMAS M., INEX : Initiative for the Evaluation of XML Retrieval, *Proceedings ACM SIGIR 2002 Workshop on XML and Information Retrieval*, 2002.

[JAA 98] JAAKKOLA T., HAUSSLER D., Exploiting Generative Models in Discriminative Classifiers, *Proceedings NIPS*, 1998.

[LEE 02] LEE M. L., YANG L. H., HSU W.YANG X., XClust: Clustering XML Schemas for Effective Integration, *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, 292–299.

[NIE 02] NIERMAN A.JAGADISH H. V., Evaluating Structural Similarity in XML Documents, *Proceedings of WebDB 2002*, 2002.

[PAP 00] PAPAKONSTANTINOU Y.VIANU V., DTD inference for views of XML data, *Proceedings of PODS'00*, 2000, 35–46.

[TER 02] TERMIER A., ROUSSET M.-C.SEBAG M., TreeFinder: a First Step towards XML Data Mining, *Proceedings of ICDM'02*, 2002.

[ZHA 01] ZHAO Y.KARYPIS G., Criterion functions for document clustering: Experiments and analysis, , 2001, Department of Computer Science, University of Minnesota, Minneapolis.