HAL
open science

Réseaux bayésiens jumelés et noyau de Fisher pondéré pour la classification de documents XML

Ait Ali Yahia Yassine, Amrouche Karima

To cite this version:

Ait Ali Yahia Yassine, Amrouche Karima. Réseaux bayésiens jumelés et noyau de Fisher pondéré pour la classification de documents XML. Revue Africaine de Recherche en Informatique et Mathématiques Appliquées, 2014, Volume 17 - 2014 - Special issue CARI'12, pp.141-154. 10.46298/arima.1971 . hal-01300057

HAL Id: hal-01300057

https://inria.hal.science/hal-01300057v1

Submitted on 8 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Special issue CARI'12

Réseaux bayésiens jumelés et noyau de Fisher pondéré pour la classification de documents XML

Ait Ali Yahia Yassine et Amrouche Karima

Ecole Nationale Supérieure d'Informatique
BP 68M Oued Smar
Oued Smart 16270 Alger
ALGERIE

y_ait_ali_yahia@esi.dz, k_amrouche@esi.dz

RÉSUMÉ. Dans le cadre de cet article, nous nous intéressons à la classification supervisée de documents structurés de type XML. Nous présentons tout d'abord un modèle génératif arborescent jumelé, basé sur le formalisme des réseaux bayésiens, afin de modéliser les documents structurés qui permet de prendre en compte simultanément l'information de contenu et l'information de structure. Ensuite nous appliquons une variante du noyau de Fisher, basée sur la pondération des composantes du vecteur de Fisher pour obtenir un modèle discriminant. Enfin, nous testons les deux modèles avec et sans pondération sur un corpus de documents XML en utilisant les méthodes CBS et SVM.

ABSTRACT. In this paper, we are presenting a learning model for XML document classification based on Bayesian networks. Then, we are proposing a model which simplifies the arborescent representation of the XML document that we have, named coupled model and we will see that this approach improves the response time and keeps the same performances of the classification. Then, we will study an extension of this generative model to the discriminating model thanks to the formalism of the Fisher's kernel. At last, we have applied a ponderation of the structure components of the Fisher's vector. We finish by presenting the obtained results on the XML collection by using the CBS and SVM methods.

.**MOTS-CLÉS :** Documents XML, réseaux bayésiens, noyau de Fisher, classification, modèles discriminants.

KEYWORDS: XML documents, Bayesian networks, Fisher's kernel, classification, discriminating models.

1. Introduction

Le développement du web et le nombre croissant de documents électroniques disponibles ont permis l'émergence de formats semi-structurés permettant la représentation et le stockage de documents textuels ou multimédias. Différents formats comme le HTML, le XHTML ou le XML sont aujourd'hui très populaires. Ces formats prennent en compte la structure logique des documents. Nous étudions ici le problème de la classification de documents structurés textuels de type XML. La classification supervisée de documents est une problématique générique de la recherche d'information. Elle est utile pour différentes tâches telles que le filtrage d'email ou de Spam, l'indexation de documents, l'organisation de corpus, etc [7] [16] [19].

Nous présentons, dans cet article, un modèle génératif pour la classification de documents structurés, inspiré des travaux de Denoyer et al [2][3]. Dans le travail présenté dans [2][3] les auteurs utilisent un réseau bayésien classique avec le noyau de Fisher pour la classification de documents XML. Néanmoins leur méthode présente deux inconvénients qu'on peut résumer comme suit : lors de l'apprentissage des paramètres du réseau bayésien certains paramètres sont ré estimés plusieurs fois, ce qui rend la méthode gourmande en temps de calcul ; par ailleurs le noyau de Fisher utilisé ne prend pas en compte les poids des nœuds (balises) dans les documents XML. Dans ce travail on applique la technique de « jumelage » des nœuds dans les réseaux bayésiens qui simplifie la structure arborescente du document XML, ce qui réduit le nombre de paramètres à estimer par rapport à la méthode classique. Cette technique a déjà fait l'objet d'un article précédent [1] où l'on montre qu'elle réduit le temps d'exécution de 30% par rapport aux réseaux classiques tout en préservant les mêmes performances pour la classification des documents XML en termes d'efficacité. On a appelé le modèle résultant : modèle génératif arborescent avec jumelage. Puis, à partir de ce modèle génératif simplifié, nous sommes passés à un modèle discriminant à l'aide du noyau de Fisher auquel on a appliqué une pondération pour les composantes du vecteur de Fisher en tenant compte des fréquences des nœuds dans les classes et ceci, comme on va le voir dans les résultats expérimentaux, améliore sensiblement les résultats de la classification. Pour voir l'apport de la pondération dans le modèle discriminant, nous avons implémenté la méthode CBS (Classifier-Based Search) [18] et la méthode SVM (Support Vecteur Machine). Nous présentons finalement les résultats obtenus sur un échantillon de la collection de documents XML, issue de la campagne d'évaluation INEX.

2. Modèle génératif avec jumelage pour les documents structurés.

Dans [2], les auteurs ont développé un modèle génératif pour la classification des documents structurés permettant de prendre en compte simultanément l'information dégagée par la structure du document et de son contenu textuel. Issu du formalisme des réseaux Bayésiens, ce modèle repose sur le principe suivant : l'auteur va tout d'abord décrire à priori la structure (plan) de son document puis « remplir » chacune de ces entités structurelles. Autrement dit, le texte apparaissant dans un nœud du document ne dépend que de l'entité structurelle qui le contient.

2.1. Document structuré

Nous représentons un document structuré comme un graphe orienté sans cycle (DAG pour Directed Acyclic Graph), ce qui correspond à la représentation usuelle utilisée dans les langages à base de balises (HTML et XML). Chaque nœud du graphe représente une entité structurelle (paragraphe, titre, section...) du document et chaque arc représente une relation hiérarchique entre deux entités (par exemple, un paragraphe est inclus dans une section). Ainsi un document *d* peut être vu comme un ensemble de nœuds, où chaque nœud n_i^d est composé de :

– Une *étiquette* (ou label) : un label peut être par exemple paragraphe, section, introduction etc. L'ensemble des étiquettes dépend des documents du corpus que nous traitons.

– Un *contenu* : qui est le texte associé au label du nœud n_i^d , un nœud peut ne pas avoir de contenu, auquel cas nous considérons le contenu comme vide par exemple le nœud de label « *document »,* dans la figure 1 n'a pas de texte.

2.2. Modèle génératif pour les documents structurés

Afin de rendre le modèle calculable et utiliser des estimations robustes, des hypothèses sur les dépendances conditionnelles concernant les différentes parties d'un document ont été formulées :

- Le réseau Bayésien associé à un document est calqué sur la structure arborescente de ce document. Ainsi la probabilité d'une sous partie dépend uniquement de la probabilité de la partie qui la contient dans l'arbre.

- Le processus de génération du texte dépend uniquement de l'étiquette du nœud dans lequel le texte apparaît (hypothèse sur le contenu).

- L'étiquette d'un nœud dépend uniquement de l'étiquette du nœud père et pas du texte de ce dernier. Cela induit une simplification supplémentaire par rapport à la première hypothèse (hypothèse sur la structure) [11].

Ces hypothèses peuvent être résumées par : *« la structure d'un document ne dépend pas du texte contenu dans ce document tandis que le texte du document dépend uniquement de l'unité structurelle qui le contient ».*

Plusieurs réseaux Bayésiens peuvent être associés à un document en fonction des dépendances à prendre en compte, dans notre cas nous ne nous intéressons qu'aux dépendances père-fils. Ce type de modèle, dont un exemple est donné en figure 1, est appelé modèle *parent*.

Soit $(n_1^d \ldots\ldots n_{|d|}^d)$ l'ensemble des nœuds d'un document d. On va considérer que le document structuré est modélisée par un réseau bayésien de $|d|$ variables aléatoires $n_1^d \ldots\ldots n_{|d|}^d$ qui correspondent aux nœuds du réseau (DAG) représentant le document. Les arcs du réseau seront modélisés par la fonction $pa(n_i^d)$ qui revoie le parent de la variable n_i^d dans le réseau. La structure du réseau ainsi construite vise à modéliser l'information d'inclusion entre différentes entités structurelles d'un document. La figure 1 montre un exemple du réseau bayésien associé au document XML.

Posons $n_i^d = (s_i^d, t_i^d)$, où s_i^d représente l'étiquette du $i^{éme}$ nœud du document d et t_i^d représente le contenu du $i^{éme}$ nœud du document d. Ainsi un document peut être vu comme la réalisation d'un couple de deux variables aléatoires : $d = (s^d, t^d)$, où : $s^d = (s_1^d, \ldots, s_{|d|}^d)$ et $t^d = (t_1^d, \ldots, t_{|d|}^d)$.

Un modèle ainsi construit nous permet de calculer la probabilité qu'un document ait été généré par une classe : $P(d/\Theta^c)$. Nous allons présenter la décomposition de cette probabilité pour le cas général c'est-à-dire $\Theta^c = \Theta$. La probabilité qu'un document ait été généré par un modèle de paramètres Θ est :

$P(d|\Theta) = P(s^d, t^d|\Theta)$.

Sous les hypothèses posées ci-dessus, cette équation peut être réécrite comme suit :

$$P(d|\Theta) = P(s^d, t^d|\Theta) = P(t^d|s^d, \Theta) \times P(s^d|\Theta) \quad (1)$$

$P(s^d|\Theta)$ est appelée probabilité structurelle et $P(t^d|s^d, \Theta)$ est appelée probabilité textuelle du document ou probabilité du contenu.

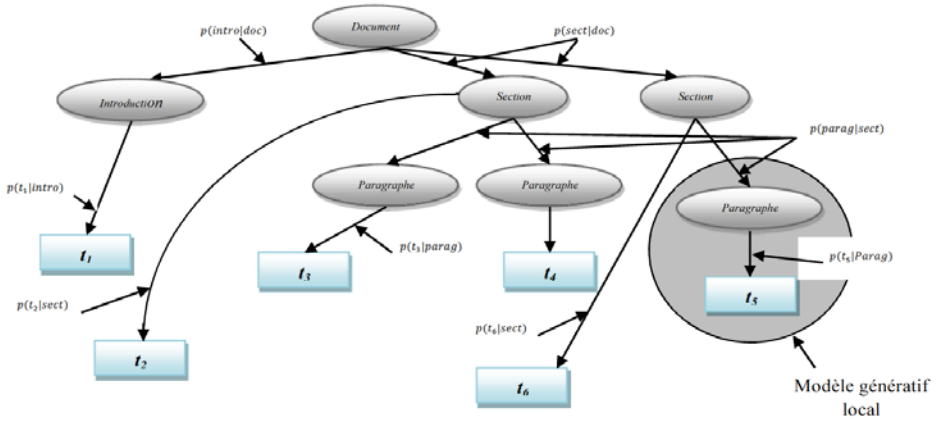

Figure1. *Le réseau Bayésien construit pour un document avec le modèle « parent ».*

2.2.1 Probabilité structurelle

La figure 1 décrit le document XML sous forme de réseau bayésien qui montre les dépendances entre les différentes composantes de structure du document.

En appliquant les hypothèses sur la structure de la section 2.2, on aura :

$$P(s^d|\Theta) = \prod_{i=1}^{|d|} P(s_i^d|pa(s_i^d), \Theta^s) \quad (2)$$

Où les paramètres Θ^s correspondent à l'estimation des probabilités $P(s_i^d|pa(s_i^d))$ qui sont les probabilités d'observer un nœud s_i^d sachant que son père est $pa(s_i^d)$.

2.2.2 Probabilité de contenu

Sous les hypothèses sur le contenu décrites dans la section 2.2, la probabilité du contenu peut être réécrite comme suit :

$$P(t^d|s^d, \Theta) = p(t_i^d \mid s^d, \Theta^t) = \prod_{i=1}^{|d|} P(t_i^d \mid s_i^d, \Theta^t)$$

Où les paramètres Θ^t correspondent à l'estimation des probabilités $P(t_i^d \mid s_i^d)$ qui sont les probabilités d'observer un texte t_i^d sachant son père (i.e. le nœud structurel de label s_i^d qui le contient).

La probabilité de contenu du nœud ne dépend que de son étiquette (label). L'idée de cette modélisation est de considérer que, pour chaque étiquette (label) possible, nous avons un modèle génératif, appelé modèle génératif local (voir figure 1), qui permet la modélisation du contenu des nœuds portant cette étiquette. C'est-à-dire :

$\Theta^t = \bigcup_{l \in \Lambda} \Theta_l^t$.

Ce qui nous donne :

$$P(t^d|s^d, \Theta^t) = \prod_{i=1}^{|d|} P(t_i^d|s_i^d, \Theta_{s_i^d}^t) . \quad (3)$$

Où $P(t_i^d|s_i^d, \Theta_{s_i^d}^t)$ est la probabilité que le contenu t_i^d ait été généré par le modèle génératif local associé au label s_i^d.

2.2.3 Probabilité finale

Les hypothèses sur la probabilité structurelle et la probabilité de contenu nous permettent donc de construire un modèle génératif de documents structurés arborescents final suivant :

$$P(d|\Theta) = P(s^d|\Theta^s) \times \prod_{i=1}^{|d|} P\left(t_i^d \middle| s_i^d, \Theta_{s_i^d}^t\right) .$$

En remplaçant la probabilité structurelle par sa formule (2), la probabilité finale s'écrit alors :

$$P(d|\Theta) = \left(\prod_{i=1}^{|d|} P(s_i^d|pa(s_i^d), \Theta^s)\right) \times \left(\prod_{i=1}^{|d|} P(t_i^d|s_i^d, \Theta_{s_i^d}^t)\right) \quad (4)$$

Avec $\Theta = \Theta^s \cup \Theta^t$ et $\Theta^t = \bigcup_{l \in \Lambda} \Theta_l^t$.

2.3. Apprentissage des paramètres

2.3.1. Apprentissage des paramètres de structure

Posons $\theta_{n,m}^s$ l'estimation de la probabilité $P(s_i^d = n|pa(s_i^d) = m, \Theta^s)$.

Notre but est donc d'estimer la probabilité $\theta_{n,m}^s$ pour chaque valeur $(n, m) \in \Lambda^2$, pour cela nous allons utiliser l'approche du maximum de vraisemblance qui consiste à estimer la probabilité d'un événement par la fréquence d'apparition de l'événement dans la base de données [17]. Cette approche nous donne alors :

$$\forall\, n, m \in \Lambda^2, \theta_{n,m}^s = \frac{\sum_{d \in D_{TRAIN}} NS_{n,m}^d + 1}{\sum_{d \in D_{TRAIN}} \sum_{n' \in \Lambda} NS_{n',m}^d + |\Lambda|} \quad (5)$$

$NS_{n,m}^{d}$ est le nombre de fois qu'un nœud de label n possède un père de label m dans l'ensemble des documents de la base d'apprentissage D_{TRAIN}. Λ est l'ensemble des étiquettes possibles (i.e. $s_i^d \in \Lambda$).

2.3.2. Apprentissage des paramètres de contenu

2.3.2.1. Modèle génératif local de type Naïve Bayes pour le contenu

L'information de contenu d'un nœud n_i^d est une information textuelle de la forme $t_i^d = \{w_{i,k}^d\}, k \in [1, \dots, |t_i^d|]$, Où $w_{i,k}^d$ est le $k^{ème}$ *mot* du nœud $\boldsymbol{i}$ du document d. L'aspect naïf de ce modèle réside dans le fait qu'on suppose l'indépendance entre les différents mots d'un contenu d'un nœud, cette indépendance nous permet de décomposer la probabilité de contenu [5] [14].

$$P(t^d|s^d, \Theta^t) = \prod_{i=1}^{|d|} P\left(t_i^d \middle| s_i^d, \Theta_{s_i^d}^t\right) = \prod_{i=1}^{|d|} \left(\prod_{k=1}^{|t_i^d|} P(w_{i,k}^d|s_i^d, \Theta_{s_i^d}^t)\right) \quad (6)$$

2.3.2.2. Apprentissage du modèle Naïve Bayes pour le modèle structuré

Comme précédemment, nous allons utiliser la méthode du *maximum de vraisemblance.* Notons $\theta_{w,l}$ l'estimation de la probabilité $P(w_{i,k}^d = w|s_i^d = l, \Theta_l^t)$.

L'estimation de cette probabilité est donnée par la formule suivante :

$$\theta_{w,l} = \frac{\sum_{d \in D_{TRAIN}} NW_{w,l}^d + 1}{\left(\sum_{d \in D_{TRAIN}} \sum_{w' \in V} NW_{w',l}^d\right) + |V|} \quad (7)$$

$NW_{w,l}^d$ est le nombre de fois que le mot w apparaît dans le document d dans un nœud de label l. V l'ensemble de mots possibles (i.e. $w \in V$), appelé vocabulaire.

2.4. Jumelage

Dans le modèle arborescent de base présenté dans la Figure 2. a), les probabilités $P(l_i|pa(l_i), \Theta^s)$ sont estimées pour tous les nœuds de l'arbre. Si un document d contient k fois une relation « *nœud-Père(nœud)* », les probabilités de ces relations sont calculées séparément les unes des autres bien qu'elles ont la même probabilité, on calculera donc k fois la même probabilité. On a remédié à ce problème en proposant une nouvelle représentation du document donnée par la Figure 2. b), que nous avons appelé *modèle jumelé*. Chaque relation « *nœud-Père(nœud)* » est représentée une seule fois en lui associant un poids qui est la fréquence d'apparition de cette relation dans l'arbre initial. Avec cette nouvelle représentation du document structuré, si un document d contient k fois un nœud « *Nœud* » de père « *pére-Nœud* », la probabilité est calculée une seule fois ensuite elle est élevée à la puissance k. Ainsi, le calcul de la probabilité structurelle se fait selon la formule suivante :

$$P(s^d|\Theta) = \prod_{i=1}^{q} \left[P(s_i^d|pa(s_i^d), \Theta^s)\right]^{freq(s_i^d - pa(s_i^d))} \quad \text{avec } q < |d| \quad (8)$$

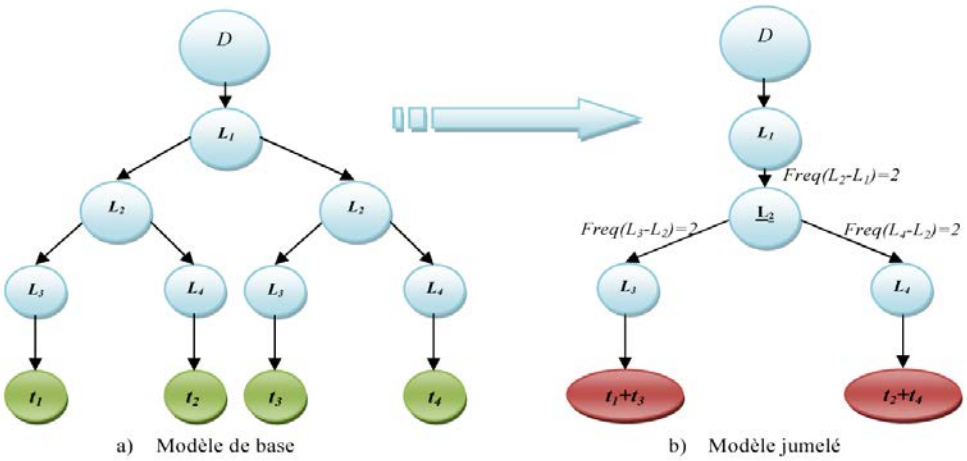

Figure 2. *Jumelage des nœuds de la structure arborescente*.

Le jumelage des nœuds améliore d'une façon significative le temps de réponse du système (environ 30%). Le temps de réponse diminue avec l'augmentation de la fréquence des nœuds dans le document, Cela est due au fait que le modèle arborescent avec jumelage calcule les probabilités des nœuds identiques qui ont des pères identiques une seule fois [1].

3. Passage au modèle discriminant par le noyau de Fisher

3.1. Le noyau de Fisher :

La méthode du noyau de Fisher a été développée au départ pour la classification de séquences biologiques par Jaakkola ([8], [9]). Son but est de transformer un modèle génératif en un 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 (CBS et SVM dans notre travail).

Pour capturer le processus génératif dans une métrique entre exemples, Jaakkola a utilisé l'espace gradient du modèle génératif. Le gradient de la log-vraisemblance, par rapport aux paramètres du modèle, décrit combien chacun des paramètres contribue au

processus de génération des exemples. Cet espace gradient préserve naturellement toutes les hypothèses structurelles que le modèle encode à propos du processus de génération [8].

Soit un modèle génératif$P(d/\Theta)$, Jaakkola propose, dans [8], de calculer le score de Fisher du document *d* comme suit :

$$U_d = \nabla_\Theta \log P(d/\Theta) . \quad (9)$$

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, Jaakkola a défini, dans [9], une similarité entre deux documents d_1 et d_2 à l'aide du noyau de Fisher suivant [14]:

$$k(d1, d2) = U_{d_1}^T \; U_{d_2} \quad (10)$$

Qui est le produit scalaire euclidien entre les vecteurs U_{d_1} et U_{d_2}. Ce noyau simplifié est alors un noyau linéaire entre les deux représentations vectorielles des documents.

Notons que les vecteurs représentant les documents peuvent aussi être utilisés avec des méthodes vectorielles classiques comme CBS, SVM, l'algorithme K-plus proches voisins, les réseaux de neurones, etc.

3.2. Application du noyau de Fisher à notre modèle génératif :

En appliquant le score de Fisher sur le modèle *« parent »*, le vecteur de Fisher s'écrit:

$$U_d = \nabla_\Theta log\, P(s^d|\Theta^s) + \sum_{l \in \Lambda} \nabla_\Theta \left(\sum_{\substack{i \\ s_i^d = l}} \log P(t_i^d |\, s_i^d, \Theta_l^t) \right)$$

$$= \nabla_\Theta \sum_{i=1}^{|d|} log\, \theta^s_{s_i^d, label(parent(n_i^d))} + \nabla_\Theta \sum_{i=1}^{|d|} \sum_{k=1}^{|t_i^d|} log\, \theta_{w_{i,k}^d, s_i^d} .$$

La représentation vectorielle d'un document structuré sera alors la composition 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. : $\sum_{i/s_i^d=l} log\, P(t_i^d | s_i^d, \Theta_l^t)$)

Après calculs, le score de Fisher pour le modèle parent est donné par le vecteur ci-dessous (figure 3) :

$$\left(\frac{NS_{l_1,l_1}^d}{\theta_{l_1,l_1}^s}, \dots, \frac{NS_{l_{|\Lambda|},l_1}^d}{\theta_{l_{|\Lambda|},l_1}^s}, \frac{NS_{l_1,l_2}^d}{\theta_{l_1,l_2}^s}, \dots, \frac{NS_{l_{|\Lambda|},l_{|\Lambda|}}^d}{\theta_{l_{|\Lambda|},l_{|\Lambda|}}^s} \vdots \frac{NW_{w_1,l_1}^d}{\theta_{w_1,l_1}^t}, \dots, \frac{NW_{w_{|V|},l_1}^d}{\theta_{w_{|V|},l_1}^t}, \frac{NW_{w_1,\lambda l_{|\Lambda|}}^d}{\theta_{w_1,l_{|\Lambda|}}^t}, \dots, \frac{NW_{w_{|V|},l_{|\Lambda|}}^d}{\theta_{w_{|V|},l_{|\Lambda|}}^t}\right)$$

Sous-vecteur correspondant au gradient sur le modèle de structure

Sous-vecteur correspondant au gradient pour les nœuds de label l_1

Sous-vecteur correspondant au gradient pour les nœuds de label $l_{|\Lambda|}$

Figure 3. *Vecteur de Fisher final obtenu pour un document d.*

Le vecteur obtenu est une composition d'un vecteur représentant l'information de structure (partie gauche) et d'un ensemble de vecteurs représentant les contenus pour les différentes étiquettes possibles des nœuds.

– $NW_{w,l}^d$ est le nombre de fois que le mot w apparaît dans le document d d'entraînement pour le nœud de label l.

– $NS_{n,m}^d$ est le nombre de fois qu'un nœud de label n possède un père de label m dans le document d.

– $\Theta_{n,m}^s$ représente la probabilité structurelle donnée par la formule (5)

– $\Theta_{w,l}^t$ représente la probabilité du contenu donnée par la formule (7)

3.3. Pondération des composantes du vecteur de Fisher

Comme les composantes du vecteur de Fisher (voir Figure 3.) sont les différentes combinaisons possibles *nœud-parent (nœud)* et *nœud-terme*, l'idée est de pondérer chaque relation comme suit :

Chaque composante structurelle correspondant à une relation *nœud-parent(nœud)* sera pondérée par le nombre d'apparition du nœud parent *« parent(nœud) »* dans la collection. Chaque composante de contenu correspondant à une relation *nœud-terme* sera pondérée par le nombre d'apparition du nœud *« nœud »* qui contient le terme dans la collection. Soit k_{l_i} le nombre d'apparition du nœud de label l_i dans la collection. Une telle pondération nous donne un vecteur de la forme suivante (Figure 4.) :

$$\left(\boldsymbol{k_{l_1}}\frac{NS^d_{l_1,l_1}}{\theta^s_{l_1,l_1}},\dots,\boldsymbol{k_{l_1}}\frac{NS^d_{l_{|\Lambda|},l_1}}{\theta^s_{l_{|\Lambda|},l_1}},\dots,\boldsymbol{k_{l_{|\Lambda|}}}\frac{NS^d_{l_1,l_{|\Lambda|}}}{\theta^s_{l_1,l_{|\Lambda|}}},\dots,\boldsymbol{k_{l_{|\Lambda|}}}\frac{NS^d_{l_{|\Lambda|},l_{|\Lambda|}}}{\theta^s_{l_{|\Lambda|},l_{|\Lambda|}}},\right.$$

$$\left.\boldsymbol{k_{l_1}}\frac{NW^d_{w_1,l_1}}{\theta^t_{w_1,l_1}},\dots,\boldsymbol{k_{l_1}}\frac{NW^d_{w_{|V|},l_1}}{\theta^t_{w_{|V|},l_1}},\dots,\boldsymbol{k_{l_{|\Lambda|}}}\frac{NW^d_{w_1,\lambda l_{|\Lambda|}}}{\theta^t_{w_1,l_{|\Lambda|}}},\dots,\boldsymbol{k_{l_{|\Lambda|}}}\frac{NW^d_{w_{|V|},l_{|\Lambda|}}}{\theta^t_{w_{|V|},l_{|\Lambda|}}}\right)$$

Figure 4. *Vecteur de Fisher final obtenu pour un document d avec pondération.*

4. Tests et résultats

Dans ce qui suit, nous allons présenter une série d'expériences pour tester notre modèle. Ces expériences ont été réalisées sur différentes collections de documents XML prises de la campagne d'évaluation INEX [6]. Afin de mettre en évidence l'apport de la pondération, nous avons implémenté, pour la classification, les méthodes SVM (Support Vecteur Machine) et CBS (Classifier-Based Search) suite au passage au modèle discriminant par le noyau de Fisher. On a construit un échantillon extrait d'INEX, pour effectuer les différents tests. Le système calcule la F-mesure micro moyenne et macro-moyenne ainsi que le rappel macro et micro moyenne en prenant en compte le type de méthode (CBS et SVM), les résultats obtenus sont résumés dans le tableau 1.

On voit à partir du tableau 1 de résultats que la pondération des composantes des vecteurs de Fisher nous permet une augmentation de 4% des mesures de performances. La figure 5 montre sur un graphe l'apport de la pondération sur la F-mesure micro et macro moyenne. De plus, les résultats montrent que la méthode CBS permet d'obtenir de meilleures performances que la méthode SVM.

	Avec Pondération		***Sans pondération***	
	CBS	***SVM***	***CBS***	***SVM***
F-Mesure micro moyenne	0,9181	0,903903	0,863344	0 ,87546
F-Mesure macro moyenne	0,9141	0,899777	0,877536	0,865585
Rappel micro moyenne	0,9189	0,905405	0,884918	0,870094
Rappel macro moyenne	0,9047	0,888888	0,86766	0,853332

Tableau 1. *Résultats obtenus pour les différents tests avec et sans pondération.*

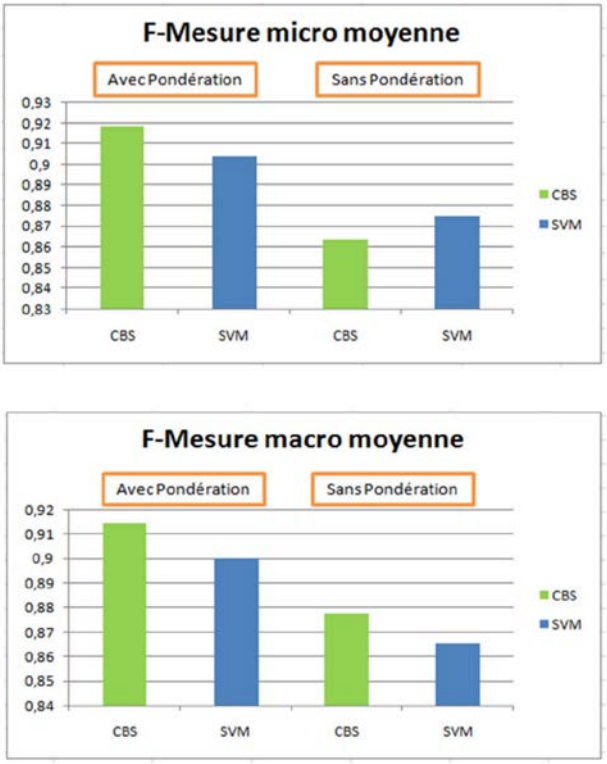

Figure 5. *F-mesure Micro et Macro moyenne avec et sans pondération***.**

5. Conclusion

Le travail présenté dans cet article, s'inscrit dans le domaine de la recherche d'information semi-structurée et plus précisément la classification supervisée de documents XML. Notre objectif était l'amélioration d'un système de classification automatique de documents XML à l'aide d'un modèle d'apprentissage discriminant basé sur les réseaux bayésiens et le noyau de Fisher proposé par [2], [3] en termes de temps d'exécution et surtout sur le plan de performance de classification.

Afin d'aboutir à cet objectif, nous avons implémenté le modèle génératif arborescent, issu du formalisme des réseaux bayésiens, capable de prendre en compte simultanément l'information de structure et l'information de contenu des documents

semi- structurés. Puis, nous avons appliqué la technique du jumelage qui nous permet de réduire le temps de réponse du système de classification tout en gardant les mêmes performances. Ce modèle est nommé modèle génératif arborescent jumelé. Et enfin nous avons présenté la technique du noyau de Fisher qui nous permet de passer de ce modèle génératif jumelé au modèle discriminant et à laquelle on a appliqué une technique de pondération.

Notre principal objectif est de comparer entre les deux modèles discriminants, obtenus à l'aide du noyau de Fisher à partir du modèle génératif jumelé, avec et sans pondération des composantes du score de Fisher. On a montré que la pondération nous permet d'obtenir de meilleures performances.

6. Bibliographie

[1] Amrouche K. and Ait Ali Yahia Y., Nodes coupling in a Bayesian Network for the automatic classification of XML documents, *International Conference on Machine and Web Intelligence*, *ICMWI'2010*, Algiers.

[2] Denoyer, L., Gallinari, P., Bayesian Network Model for semi-structured document classification, IP&M *Bayesian Network and Information Retrieval*, 1-25, 2004

[3] Denoyer, L., Gallinari, P., Un modèle de mixture de modèles génératifs pour les documents structurés multimédias, DN-8/2004, *fouille de textes*, pages 35 à 54.

[4] A. Doucet and H. Ahonen-Myka, "Naïve clustering of a large XML document collection", In proceedings of the First Annual Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), pages 81-88, 2002

[5] S. Eyheramendy, D. D. Lewis and D. Madigan, "On the Naïve Bayes model for text categorization", In proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003.

[6] N. Fuhr, N. Govert , G. Kazai and M. Lalmas, "INEX: Initiative for the Evaluation of XML", in Proceedings ACM SIGIR 2002 Workshop on XML and Information Retrieval, 2002.

[7] M. Granitzer, Hierarchical text classication using methods from machine learning, Master's Thesis at Graz University of Technology, 2003.

[8] Jaakkola T., Haussler D., Exploiting generative models in discriminative classifiers, *Advanced in Neural information processing systems,* 11, 1998.

[9] Jaakkola T., Diekhans M. and Haussler D.,Using the Fisher kernel method to detect remote protein homologies, *Intelligent Systems for Molecular Biology Conference* (*ISMB'99*), Heidelberg, Germany, AAAI.

[10] F. V. Jensen, An introduction to Bayesian Netwroks, UCL Press 1996

[11] D. D. Lewis, "Naïve (Bayes) at forty: the independence assumption in information retrieval", in: proceedings of ECML-98, Springer Verlag, Heidelberg, Chemnitz, DE, pp4-15.

[12] H. Lounis, "Apprentissage bayésien: séminaire sur l'apprentissage automatique", programme de doctorat en informatique cognitive, université du Québec à Montréal, 2006

[13] J. Pearl, "From Bayesian Networks to causal networks", In 49th Session of the International Statistical Institute, 1993

[14] B. Piwowarski, G Faure and P Gallinari, "Bayesian networks and INEX, In proccedings of the first annual workshop of the Initiative for the Evaluation of XML retrieval (INEX), DELOS workshop, ERCIM Dagstuhl, Germany, 2002.

[15] B. Piwowarski and P. Gallinari, "A Bayesian network model for page retrieval in a hierarchically structured collection", in XML Workshop of the 25th ACM SIGIR Conference, Tampere, Finlande 2002.

[16] C. Y. Quek, "Classification of world wide web documents", school of computer science, Carnegie Mellon university, Senior Honor Thesis, 1997.

[17] M. Ramoni and P. Sebastiani, Parameter stimation in Bayesian Networks, Technical Report KMI-TR-57, Knowledge Media Institute, UK 1997

[18] Schühmacher J.P.C., Classifier-Based search in large document collections, Master's thesis 2011.

[19] J. Yi and N. Sundaresan, "A classifier for semi-structured documents", In Proceedings of the 6th ACM SIGKDD, pages 640-644, 2000.