Apprentissage semi–supervisé de fonctions d'ordonnancement

Vinh Truong, Massih-Reza Amini

Laboratoire d'Informatique de Paris 6
104, Avenue du Président Kennedy, 75016 Paris, France
{truong, amini}@poleia.lip6.fr
http://www-connex.lip6.fr

Résumé. Nous présentons dans cet article un algorithme inductif semi-supervisé pour la tâche d'ordonnancement bipartite. Les algorithmes semi–supervisés proposés jusqu'à maintenant ont été étudiés dans le cadre strict de la classification. Récemment des travaux ont été réalisés dans le cadre transductif pour étendre les modèles existants en classification au cadre d'ordonnancement. L'originalité de notre approche est qu'elle est capable d'inférer un ordre sur une base test non–utilisée pendant la phase d'apprentissage, ce qui la rend plus générique qu'une méthode transductive pure. Les résultats empiriques sur la base CACM contenant les titres et les résumés du journal *Communications of the Association for Computer Machinery* montrent que les données non–étiquetées sont bénéfiques pour l'apprentissage de fonctions d'ordonnancement.

1 Introduction

Avec le développement des bibliothèques électroniques, il est devenu nécessaire de concevoir des méthodes automatiques pour la recherche de données pertinentes par rapport à une requête donnée. Pour de telles applications, il s'agit plus d'ordonner les exemples que de les discriminer.

La communauté d'apprentissage a formulé cette problématique à travers le nouveau paradigme d'apprentissage supervisé de fonctions d'ordonnancement. Dans ce cas, il s'agit d'apprendre une correspondance entre un ensemble d'instances et un ensemble d'alternatives capable d'ordonner les alternatives par rapport à une instance donnée. Par exemple, dans le cas de la recherche documentaire (RD), une instance représente une requête et les alternatives sont les documents concernés par cette requête et le but est d'inférer un ordre partiel sur l'ensemble des alternatives de façon à ce que les documents pertinents par rapport à la requête soient mieux ordonnés que les documents non-pertinents.

Dans ce papier, nous nous plaçons dans le cadre de l'ordonnancement bipartite dans lequel les instances sont soit positives soit négatives et où il s'agit d'ordonner les instances positives au–dessus des instances négatives. Ce cadre restreint englobe de nombreuses applications de la recherche d'information telle que le résumé automatique (Amini et al., 2005) ou la recherche

de passages pertinents dans les systèmes de questions/réponses (Usunier et al., 2005) et a récemment fait l'objet de plusieurs études aussi bien sur un plan pratique que théorique (Agarwal et Roth (2005); Rudin et al. (2005); Freund et al. (2003)).

Le principal inconvénient de l'apprentissage supervisé de fonctions d'ordonnancement est que l'étiquetage des instances nécessite l'intervention d'un expert qui doit examiner manuellement une grande quantité de données. Dans le cadre de la classification, la communauté d'apprentissage s'est intéressée depuis la fin des années 90 au problème d'apprentissage semi–supervisé qui consiste à prendre en compte les données étiquetées et non–étiquetées dans le processus d'apprentissage.

L'originalité de notre approche est que nous proposons un algorithme d'apprentissage semi-supervisé pour la tâche d'ordonnancement bipartite. La plupart des algorithmes d'ordonnancement semi-supervisés sont des techniques transductives à base de graphes, qui permettent d'étiqueter les exemples non-étiquetés d'une base test fixe. Nous préconisons une approche inductive à ce problème où il s'agit d'apprendre une fonction d'ordonnancement à partir de deux bases d'apprentissage, étiquetée et non-étiquetée, et qui est capable d'ordonner de nouveaux exemples qui n'ont pas été utilisés pour entraîner le modèle. Notre algorithme adopte une approche itérative en initialisant d'abord une fonction d'ordonnancement à partir des exemples étiquetés de la base d'apprentissage et en apprenant la structure des données non-étiquetées de la base d'apprentissage par une méthode transductive. Il répète ensuite deux étapes jusqu'à ce que les critères de convergence ou d'arrêt soient atteints. La première étape consiste à ordonner un sous-ensemble d'exemples non–étiquetés avec la sortie de la fonction d'ordonnancement et ensuite à calculer une dissimilarité entre cet ordre et celui inféré par la méthode transductive sur ce sous-ensemble. Dans la deuxième étape, l'algorithme apprend une nouvelle fonction d'ordonnancement à partir de l'ensemble des données étiquetées et du sous-ensemble d'exemples non–étiquetés trouvé à l'étape précédente. Nous montrons l'efficacité de cette approche pour la tâche RD.

Dans ce qui suit, nous reviendrons en section 2 à la tâche d'ordonnancement bipartite dans le cas supervisé. Dans la section 3 nous présenterons notre algorithme d'ordonnancement semi–supervisé et dans la section 4, nous présenterons les résultats obtenus sur la base CACM[1] constituée des titres et des résumés du journal *Communications of the Association for Computer Machinery*. Finalement nous discuterons des résultats obtenus en section 4.2.

2 La tâche d'ordonnancement bipartite

2.1 Le cadre supervisé

On peut formaliser le problème d'ordonnancement bipartite comme suit. On considère une base d'apprentissage $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_{-1})$ constituée d'un ensemble d'exemples positifs $\mathcal{S}_1$ et négatifs $\mathcal{S}_{-1}$. Les exemples $x \in \mathcal{S}_1$ et $x' \in \mathcal{S}_{-1}$ sont caractérisés dans un espace $\mathcal{X}$. Le but

[1] http ://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cacm/

est alors d'apprendre une fonction $h : \mathcal{X} \rightarrow \mathbb{R}$ qui donne des scores plus élevés aux exemples positifs qu'aux exemples négatifs. Formellement,

$$\forall (x, x') \in \mathcal{S}_1 \times \mathcal{S}_{-1}, h(x) > h(x')$$

L'hypothèse de base ici est que les exemples positifs et négatifs sont respectivement échantillonnés i.i.d suivant les distributions inconnues $\mathcal{D}_1$ et $\mathcal{D}_{-1}$ sur $\mathcal{X}$. Le risque d'une fonction score h est alors mesuré par son *erreur moyenne d'ordonnancement* suivant $\mathcal{D}_1$ et $\mathcal{D}_{-1}$:

$$\mathcal{R}_{\mathcal{D}_1,\mathcal{D}_{-1}}(h) = \mathbb{E}_{x\sim D_1, x'\sim D_{-1}}[[h(x) < h(x')]]$$

Où $[[pr]]$ est la fonction indicatrice valant 1 si le prédicat pr est vrai et 0 sinon. L'erreur moyenne d'ordonnancement $\mathcal{R}_{\mathcal{D}_1,\mathcal{D}_{-1}}(h)$ est la probabilité qu'un exemple positif échantillonné aléatoirement suivant $\mathcal{D}_1$ ait un score plus faible qu'un exemple négatif échantillonné aléatoirement suivant $\mathcal{D}_{-1}$ (Cortes et Mohri, 2003). L'erreur empirique d'ordonnancement correspondante de h sur une base d'apprentissage $\mathcal{S} = (S_1, S_{-1})$ est (Freund et al. (2003))

$$\hat{\mathcal{R}}_{(S_1,S_{-1})}(\mathcal{S}, h) = \frac{1}{|\mathcal{S}_1||\mathcal{S}_{-1}|} \sum_{x\in\mathcal{S}_1} \sum_{x'\in\mathcal{S}_{-1}} [[h(x) < h(x')]] \tag{1}$$

Le coût d'ordonnancement (1) est une fonction non convexe et son optimisation est non–triviale. Pour apprendre h, une approche couramment utilisée est de borner $[[h(x) < h(x')]]$ par une fonction convexe $C_g(x, x', h)$ et d'optimiser le coût suivant :

$$L_g(\mathcal{S}, h) = \frac{1}{|\mathcal{S}_1|.|\mathcal{S}_{-1}|} \sum_{x\in\mathcal{S}_1} \sum_{x'\in\mathcal{S}_{-1}} C_g(x, x', h) \tag{2}$$

Il a été démontré que dans le cas où $\lim_{(x,x')\rightarrow(0,0)} C_g(x, x', h) = 1$, minimiser (2) revient à minimiser (1) (Bartlett et Long, 1998; Clémençon et al., 2005).
Nous allons présenter dans la section suivante l'algorithme supervisé *LinearRank* optimisant le critère (2). Cet algorithme a été appliqué avec succès à la tâche de résumé automatique de textes (Amini et al., 2005).

2.2 L'algorithme supervisé *LinearRank*

Nous cherchons ici à apprendre les poids $B = (\beta_i)$ d'une combinaison linéaire des entrées, $h(x) = \sum_{i=1} \beta_i x_i$, en prenant comme borne supérieure de $[[h(x) < h(x')]]$ la fonction exponentielle $C_g(x, x', h) = e^{-(h(x)-h(x'))}$. L'apprentissage de la fonction score h revient alors à trouver les poids $B = (\beta_i)$ qui optimisent le critère

$$L_{exp} = \frac{1}{|\mathcal{S}_1|.|\mathcal{S}_{-1}|} \left(\sum_{x\in\mathcal{S}_1} \sum_{x'\in\mathcal{S}_{-1}} e^{-\sum_{i=1}^{d} \beta_i (x_i - x'_i)} \right) \tag{3}$$

Un avantage d'utiliser un coût exponentiel pour $C_g(x, x', h)$ et une fonction score linéaire est que le critère (3) peut se calculer avec une complexité linéaire par rapport au nombre

d'exemples. En effet, le critère L_{exp} peut s'écrire dans ce cas comme suit :

$$L_{exp}(\mathcal{S}, \beta) = \frac{1}{|S_1||S_{-1}|} \left(\sum_{s \in \mathcal{S}_1} e^{-\sum_{i=1}^{d} \beta_i s_i} \right) \left(\sum_{s \in \mathcal{S}_{-1}} e^{\sum_{i=1}^{d} \beta_i s_i} \right) \quad (4)$$

Un autre intérêt de la fonction de coût exponentiel est que des algorithmes d'optimisation standard permettent d'effectuer sa minimisation. Dans notre cas nous avons utilisé l'algorithme *LinearRank* (Amini et al., 2005) qui est une adaptation de l'algorithme *iterative scaling* développé pour la classification par (Lebanon et Lafferty, 2001).

3 La tâche d'ordonnancement semi–supervisée

3.1 Notation

Dans le formalisme de la tâche d'ordonnancement semi–supervisé, nous supposons que l'ensemble d'apprentissage est constitué d'instances étiquetées dont l'étiquette reflète un jugement de pertinence ainsi qu'un ensemble d'instances non étiquetées. Il s'agit ainsi d'apprendre une fonction score à partir d'une base d'instances étiquetées $\mathcal{S}_L = \{x_i, y_i\}_{i=1}^{n_L}$ et d'une base d'instance non–étiquetées $\mathcal{S}_u$, de taille n_u, où les instances $x \in \mathcal{X} \subset \mathbb{R}^d$ sont décrites par des vecteurs de dimension d.

3.2 Méthode d'ordonnancement transductive

Les méthodes semi–supervisées qui ont été proposées en ordonnancement bipartite sont basées sur une hypothèse de variétés (Zhou et al., 2004b; Chu et Ghahramani, 2005; Agarwal, 2006). Une variété peut être définie comme un espace topologique qui est localement euclidien. Par exemple, toute ligne dans un espace euclidien est une variété de dimension 1 et toute surface constitue une variété de dimension 2. Les applications des variétés sont nombreuses en mathématiques et en physiques et ont été récemment introduites en apprentissage, principalement pour la tâche de discrimination. Les méthodes utilisant la notion de variété supposent que les exemples se trouvent sur une variété de dimension inférieure à l'espace de départ et que les scores des exemples proches sur la variété sont assez similaires.

Ces algorithmes cherchent alors à exploiter la nature intrinsèque des données (c–à–d une variété) pour améliorer l'apprentissage de la fonction de décision. Par exemple, les méthodes semi–supervisées faisant l'hypothèse de variétés utilisent la grande quantité d'exemples non–étiquetés pour pouvoir estimer cette structure. Pour ce faire, un graphe incorporant l'information de voisinage local est construit avec une méthode telle que les K plus proches voisins. Les noeuds sont alors constitués des exemples étiquetés et non–étiquetés de la base d'apprentissage et les poids reflètent la similarité entre les exemples voisins. La définition de cette similarité dépend des algorithmes proposés.

Après avoir estimé la variété, la plupart de ces méthodes s'attachent à trouver les étiquettes des exemples non–étiquetés en exploitant directement le graphe en propageant par exemple les étiquettes des données étiquetées à leurs voisins non–étiquetés (Zhou et al., 2004a). Ces

algorithmes ne peuvent ainsi pas étiqueter les exemples absents de la phase d'apprentissage puisqu'ils ne font pas parti des noeuds du graphe. Ces méthodes sont dites transductives par opposition aux méthodes inductives, qui sont capables d'ordonner d'autres exemples que ceux qui ont été utilisés pour apprendre.

Récemment, des méthodes transductives à base de graphes ont été adaptées à la tâche d'ordonnancement bipartite. Par exemple, (Zhou et al., 2004b) a adapté ses travaux de classification semi–supervisée (Zhou et al., 2004a) au cas d'ordonnancement transdutif. Son algorithme construit d'abord un graphe valué et non orienté en connectant petit à petit les points les plus proches jusqu'à ce que le graphe devienne connexe. Il affecte ensuite un score pour chacune des instances, 1 pour les instances positives et 0 pour les autres. Les scores sont alors propagés à travers le graphe jusqu'à la convergence. À la fin, les scores obtenus permettent d'induire un ordre sur l'ensemble des instances non–étiquetées. L'algorithme proposé dans ce papier étant en partie basé sur cette méthode, nous allons le décrire plus en détails :

Soit $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ une métrique sur $\mathcal{X}$ et soit f la fonction score qui donne à chaque instance x_i un score noté f_i. f peut alors être vue comme un vecteur $f = [f_1, \ldots, f_{n_L+n_u}]^T$ avec $n_L + n_u$ le nombre total d'instances (étiquetées ou non) de la base d'apprentissage. On définit aussi le vecteur $y = [y_1, \ldots, y_{n_L+n_u}]^T$ tel que $y_i = 1$ si x_i est une instance positive et $y_i = 0$ pour les autres. L'algorithme proposé dans (Zhou et al., 2004b) peut alors être résumé ainsi :

Algorithme 1 : Algorithme d'ordonnancement transductif

Entrée : $\mathcal{S} = \mathcal{S}_L \cup \mathcal{S}_{unl}$

Initialisation :

- Calculer les distances $d(x_i, x_j)$ entre toutes les instances de la base d'apprentissage $\mathcal{S}$.
- Ordonner les distances $d(x_i, x_j)$ dans l'ordre décroissant.
  Connecter deux points avec un arc en suivant l'ordre obtenu jusqu'à ce que le graphe devienne connexe.
- Calculer la matrice d'affinité $W = [w_{ij}]$ définie par

$$w_{ij} = \begin{cases} \exp\left(-\frac{d(x_i,x_j)}{2\sigma 2}\right), \text{ s'il y a un arc entre } x_i \text{ et } x_j \\ 0, \text{ sinon} \end{cases}$$

- $S \leftarrow D^{-\frac{1}{2}}.W.D^{\frac{1}{2}}$ avec D la matrice diagonale telle que d_{ii} est égal à la somme des éléments de la $i^{\text{ème}}$ ligne de W.
- Construire le vecteur y telle que $y_i = 1$ si x_i est une instance positive, 0 sinon.
- $f^{(0)} \leftarrow y, t \leftarrow 0$

répéter

- $f^{(t+1)} \leftarrow \alpha S f^{(t)} + (1 - \alpha)y$
- $t \leftarrow t + 1$

jusqu'à *Convergence* ;

Sortie : Le vecteur f

Comme préconisé par (He et al., 2004), nous avons utilisé la méthode des K plus proches voisins dans la construction du graphe pour avoir plus de connexions entre les instances. En effet, la base CACM utilisée dans nos expériences comporte peu d'exemples positifs pour chaque requête. Augmenter le nombre de connexions permet ainsi d'augmenter l'influence des exemples étiquetés positifs sur les scores des exemples non–étiquetés.

3.3 Le modèle semi–supervisé inductif

La méthode supervisée *LinearRank* est une technique inductive dans le sens où, une fois le critère (3) optimisé, elle est capable d'ordonner les instances non vues durant la phase d'apprentissage. La méthode transductive quant à elle exploite la structure des données pour ordonner les instances en se basant sur leur similarité par rapport aux exemples positifs. Dans ce papier, nous nous intéressons à combiner les deux méthodes pour profiter de chacun de leurs avantages. Notre approche consiste à trouver un compromis entre optimiser le coût exponentiel (3) et respecter l'ordre trouvé à partir de la variété sur un sous–ensemble de données étiquetées.

Notre approche consiste dans un premier temps à apprendre (1) une fonction score h avec l'algorithme *LinearRank* en minimisant le coût exponentiel sur l'ensemble des données étiquetées et (2) un ordre total sur les données non–étiquetées avec la méthode transductive décrite dans la section précédente.

La partie itérative de notre algorithme répète alors deux étapes jusqu'à ce que le critère de convergence ou qu'un nombre maximum d'itérations soit atteint (algorithme 2) : la première étape consiste à sélectionner les n instances non–étiquetées les mieux ordonnées par la sortie de la fonction h. Suite à cela, nous calculons une dissimilarité entre l'ordre trouvé par la fonction h et celui trouvé par la méthode transductive. À cette étape, nous faisons l'hypothèse que l'ordre sur ce sous–ensemble trouvé par la méthode transductive est plus pertinente que celle trouvée par la fonction h : la méthode transductive exploite en effet la structure des données. Nous définissons la dissimilarité entre les deux ordres par le nombre de paires de préférence différentes :

$$d(h,\phi) = \sum_{j=1}^{n} [[h(x_{\phi(j+1)}) < h(x_{\phi(j)})]] \tag{5}$$

avec $\phi(j)$ une fonction qui retourne l'index de l'instance ordonnée au rang j par la méthode transductive. Dans une deuxième étape nous cherchons à trouver une nouvelle fonction score qui minimise le coût exponentiel régularisé $L_{exp}(S,h) + \lambda\delta_{exp}(h,\phi)$, où :

$$\delta_{exp}(h,\phi) = \sum_{j=1}^{n} e^{h(x_{\phi(j+1)}) - h(x_{\phi(j)})} \tag{6}$$

avec δ_{exp} qui est la borne supérieure de (5). Le terme de régularisation λ permet de pondérer l'apport des données non–étiquetées dans l'apprentissage et il est fixé pour toutes les itérations de notre algorithme. Nous essayons ainsi de trouver une fonction d'ordonnancement qui minimise le nombre de couples de préférence mal ordonnées et qui donne un ordre sur les instances non–étiquetées les mieux ordonnées le plus proche de celui trouvé par la méthode transductive.

L'algorithme général peut ainsi résumer par ce qui suit :

Algorithme 2 : Algorithme d'ordonnancement inductif semi–supervisé

Entrée : $\mathcal{S} = \mathcal{S}_L \cup \mathcal{S}_{unl}$

Initialisation :

- Apprendre une fonction score $h^{(0)}$ en optimisant le coût $L_{exp}(\mathcal{S}_L, h)$
- Apprendre un ordre total sur les exemples non–étiquetés avec la méthode transductive de l'algorithme 1
- $t \leftarrow 0$

répéter

- Sélectionner les exemples non–étiquetés $S_{unl}^{(t)}$ les mieux ordonnés par $h^{(t)}$.
- Produire la fonction index $\phi^{(t)}$ à partir de la méthode transductive.
- Apprendre une nouvelle fonction score en optimisant le coût exponentiel régularisé

$$h^{(t+1)} = \underset{h}{\operatorname{argmin}}\, L_{exp}(\mathcal{S}_L, h) + \lambda \sum_{j=1} e^{h^{(t)}(x_{\phi^{(t)}(j+1)}) - h^{(t)}(x_{\phi^{(t)}(j)})}$$

- $t \leftarrow t + 1$

jusqu'à *Convergence de* $L_{exp}(\mathcal{S}_L, h) + \lambda\delta_{exp}(h, \phi) \vee t \leq T$;

Sortie : La fonction h

Dans ce papier, nous avons utilisé la fonction coût exponentielle ainsi qu'une mesure de dissimilarité de même nature. D'autres fonctions de coût et de dissimilarité sont néanmoins envisageables en utilisant d'autres fonctions convexes qui bornent la fonction indicatrice (le *logit* par exemple). Un travail similaire au nôtre est celui de (Agarwal, 2006) qui a récemment proposé d'étendre les travaux de (Belkin et Niyogi, 2004) au cadre de l'ordonnancement bipartite semi–supervisé. L'étude de (Agarwal, 2006) concerne plus un cadre d'ordonnancement transductif mais l'auteur propose d'utiliser la technique développée dans (Sindhwani et al., 2005) pour rendre l'algorithme inductif. Notre algorithme est inductif dans sa construction itérative ce qui présente l'avantage d'être plus rapide à l'exécution.

4 Expériences

4.1 Base utilisée

Pour montrer de façon empirique que les instances non–étiquetées peuvent être utiles pour l'ordonnancement bipartite, nous avons comparé notre algorithme à l'algorithme supervisé *LinearRank* (Amini et al., 2005). Pour évaluer ces méthodes, nous nous sommes basés sur deux critères couramment utilisés dans la communauté de recherche d'information : l'aire sous la Courbe ROC (AUC) et la précision moyenne. Les expériences ont été menées sur la base CACM qui rassemble les titres et les résumés provenant du journal *Communications of the Association for Computer Machinery* (CACM).

Nous avons dans un premier temps prétraité les données pour chaque requête. Nous avons ainsi retiré pour chaque requête les documents pertinents ne contenant aucun mot de la requête. À partir de l'ensemble des documents obtenu, un ensemble de test a été crée aléatoirement en sélectionnant la moitié des documents. Nous avons gardé uniquement les requêtes qui contenaient suffisamment de documents pertinents dans la base d'apprentissage et de test. 12 requêtes ont été alors retenues.

Pour chaque requête, nous avons évalué les méthodes en formant cinq bases d'apprentissage et de test différentes. La méthode semi–supervisée a été appliquée sur toute la base d'apprentissage (étiquetée et non–étiquetée) en fixant un taux de données étiquetées permettant d'avoir au moins une instance positive dans la partie étiquetée. Pour connaître l'apport des données non–étiquetées, nous avons entraîné le modèle supervisé uniquement sur cet ensemble étiqueté. Les résultats obtenus en fixant les paramètres λ à 1, n et K à 10 ont été reportés dans le tableau 1. Les valeurs de test correspondent aux moyennes des résultats en AUC et en précision moyenne sur les 5 bases d'apprentissage et test considérées. Le tableau 2 donne les résultats moyennés par rapport aux requêtes.

4.2 Résultats et discussion

Les résultats obtenus sur la précision moyenne montrent que le modèle supervisé obtient de meilleurs performances que celui semi–supervisé. Ce résultat peut s'expliquer par le fait que notre algorithme n'optimise pas ce critère. Cependant le critère AUC a été de nombreuses fois utilisé en Recherche d'Information. En effet, ce critère est plus facile à optimiser que la précision moyenne et plusieurs études ont montré que ces deux critères étaient en général fortement corrélés (Caruana et Niculescu-Mizil, 2004). Optimiser le critère AUC permet ainsi d'optimiser la précision moyenne. Dans notre cas, le déséquilibre pourrait expliquer en partie les résultats que l'on obtient. En effet, pour chaque requête, il existe un nombre très limité d'exemples positifs. Pour améliorer ce critère, nous aurions pu aussi optimiser directement la précision moyenne (Metzler, 2005) ou un critère dérivé de l'AUC (Rudin et al., 2005), qui permet à l'algorithme de se concentrer sur les instances ordonnées de la liste.

Par contre, les résultats obtenus sur la mesure AUC montrent que notre méthode semi–supervisée obtient clairement de meilleures performances sur l'ensemble des requêtes. Nous notons néanmoins une baisse conséquente pour la requête 17. En regardant de plus près les résultats, nous avons remarqué que cette requête contient très peu de mots, ce qui pourrait ainsi biaiser la dissimilarité basée sur la variété. Néanmoins, nous obtenons des gains importants pour plus de la moitié des requêtes. En moyenne, l'approche semi–supervisée permet ainsi un gain d'environ $4,2\%$.

Les résultats empiriques obtenus sur l'AUC montrent des résultats plus qu'encourageant. En effet, notre méthode cherche à améliorer ce critère en utilisant des exemples non–étiquetés. Cependant, les résultats sur la précision moyenne sont plus surprenants. La baisse des performances sur ce critère que l'on observe en moyenne et pour une grande partie des requêtes tempère les résultats obtenus sur l'AUC.

	7		10		11		14	
	AUC	Prec	AUC	Prec	AUC	Prec		
Sup	81,4	21,6	80,2	20,1	98	31,5	90,8	32,6
Semi	90,8	18,9	87,9	23,8	97,9	25	93	28,8

	17		25		27		29	
	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec
Sup	95,6	35	83,2	29,4	89,7	42,9	91,4	11,3
Semi	90,3	27,4	91,2	12,3	95,8	44,4	91,7	11,6

	42		43		58		60	
	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec
Sup	72,1	31,6	98,9	49	86,2	20,2	98,8	57,1
Semi	89,6	36,9	98,4	38,9	87,3	25,6	97,7	43,5

TAB. 1 – *Performance en AUC (colonne AUC) et précision moyenne (colonne Prec) de l'algorithme* LinearRank *(ligne Sup) et semi–supervisé (ligne Semi) pour chaque requête identifiée par un entier.*

	AUC		Prec	
	moy	écart	moy	écart
Sup	88,9	+4,2	31,9	-13,5
Semi	92,6	0	28.1	0

TAB. 2 – *Moyenne des performances sur toutes les requêtes en AUC (colonne AUC) et en précision moyenne (colonne Prec) de l'algorithme* LinearRank *(ligne Sup) et semi–supervisé (ligne Semi). Pour chaque résultat, nous indiquons les moyennes (sous–colonne moy) et l'écart en pourcentage (sous–colonne écart) des algorithmes par rapport à la méthode supervisé* LinearRank

5 Conclusion

La principale contribution de ce papier est une méthode d'ordonnancement bipartite semi–supervisée inductive. Notre approche est une combinaison d'une méthode supervisée et d'une méthode transductive à base de graphe. Elle est générale dans le sens où d'autres fonctions coût que le coût exponentiel et d'autres méthodes transductives peuvent être utilisées. Les résultats obtenus sur le critère AUC montrent une amélioration significative par rapport à la méthode supervisée, tendant à montrer ainsi l'apport possible des exemples non étiquetés. Cependant, les résultats obtenus sur la précision moyenne montrent une dégradation des performances. Les deux critères étant généralement corrélés, ce résultat est assez surprenant. Pour la suite des travaux, nous allons ainsi tester notre algorithme sur d'autres bases pour confirmer les performances obtenues. Ce papier étant une première étude pour la tâche d'ordonnancement semi–supervisée, nous avons uniquement fourni une partie pratique pour la tâche plus restreinte de

l'ordonnancement bipartite. Il est néanmoins à noter que les techniques inductives d'apprentissage semi-supervisé ont été exclusivement développées dans le cadre de la classification (Amini et Gallinari, 2003) et que les résultats obtenus dans ce papier sont un bon présage quant à l'utilisation des données non–étiquetées dans d'autres cadres d'apprentissage supervisé. Une direction intéressante à explorer serait d'apprendre conjointement avec les données étiquetées et non–étiquetées pour la tâche de l'extraction d'information (Amini et al., 2000).

Remerciement

Ce travail a été partiellement financé par le programme de IST de la communauté européenne, dans le cadre du réseau d'excellence PASCAL, IST-2002-506778. Cette publication concerne uniquement les points de vue des auteurs.

Références

Agarwal, S. (2006). Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*.

Agarwal, S. et D. Roth (2005). Learnability of bipartite ranking functions. In *Proceedings of the 18th Annual Conference on Learning Theory*.

Amini, M., N. Usunier, et P. Gallinari (2005). Automatic text summarization based on word-clusters and ranking algorithms. In *European Conference on Information Retrieval (ECIR'05)*, Santiago de Compostella, pp. 142–156.

Amini, M.-R. et P. Gallinari (2003). Semi-supervised learning with explicit misclassification modeling. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003*, pp. 555–560.

Amini, M.-R., H. Zaragoza, et P. Gallinari (2000). Learning for sequence extraction tasks. In *Proceedings of the 6th Recherche d'Information Assistée par Ordinateur, RIAO 2000*, pp. 476–490. CID.

Bartlett, P. L. et P. M. Long (1998). Prediction, learning, uniform convergence, and scale-sensitive dimensions. *Journal of Computer and System Sciences 56*(2), 174–190. (special issue on COLT'95).

Belkin, M. et P. Niyogi (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning 56*(1-3), 209–239.

CACM. http ://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cacm/.

Caruana, R. et A. Niculescu-Mizil (2004). Data mining in metric space : An empirical analysis of suppervised learning performance criteria. In *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining (KDD'04)*.

Chu, W. et Z. Ghahramani (2005). Extensions of gaussian processes for ranking : semi-supervised and active learning. In *Proceedings of the NIPS'05 Workshop on Learning to Rank*.

Clémençon, S., G. Lugosi, et N. Vayatis (2005). Ranking and scoring using empirical risk minimization. In *COLT*, pp. 1–15.

Cortes, C. et M. Mohri (2003). Auc optimization vs. error rate minimization. In *NIPS*.

Freund, Y., R. Iyer, R. E. Schapire, et Y. Singer (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res. 4*, 933–969.

He, J., M. Li, H.-J. Zhang, H. Tong, et C. Zhang (2004). Manifold-ranking based image retrieval. In *MULTIMEDIA '04 : Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, pp. 9–16. ACM Press.

Lebanon, G. et J. Lafferty (2001). Boosting and maximum likelihood for exponential models. *Technical Report CMU-CS-01-144, School of Computer Science, CMU*.

Metzler, D. (2005). Direct Maximization of Rank-based Metrics. Technical report, CIIR.

Rudin, C., C. Cortes, M. Mohri, et R. E. Schapire (2005). Margin-Based Ranking Meets Boosting in the Middle. In *COLT'05*, pp. 63–78.

Sindhwani, V., P. Niyogi, et M. Belkin (2005). Beyond the point cloud : from transductive to semi-supervised learning. In *ICML*, pp. 824–831.

Usunier, N., M.-R. Amini, et P. Gallinari (2005). Combinaison de fonctions de préférence par boosting pour la recherche de passages dans les systèmes de question/réponse. In *ECG*, pp. 1–6.

Zhou, D., O. Bousquet, T. Lal, J. Weston, et B. Schölkopf (2004a). Learning with local and global consistency. Volume 16, Cambridge, MA, USA, pp. 321–328. MIT Press.

Zhou, D., J. Weston, A. Gretton, O. Bousquet, et B. Schölkopf (2004b). Ranking on data manifolds. Volume 16, Cambridge, MA, USA, pp. 169–176. MIT Press.

Summary

The growing availability of on–line resources requires the conception of generic approaches that are able to automatically find relevant entities with respect to a user's demand. Recently there has been an increasing interest of the Machine Learning community for the task of ranking by supervised learning of scoring functions. The aim is to learn a mapping from instances to rankings over a finite set of alternatives. Labeling large amounts of data may require expensive human resources, which are unfeasible in most applications. It has been shown in the classification framework that learning with both labeled and unlabeled data may lead to a more efficient decision rule than learning with labeled examples alone. In this paper, we propose a semi–supervised method for the bipartite learning task, which can rank unseen instances. We have led experiments on the real–life dataset CACM gathering titles and abstracts from the journal Communications of the Association for Computer Machinery. The empirical results have shown the potential of this approach in the context of Document Retrieval.