Vers un apprentissage multi-label rapide en grande dimension – Une étude préliminaire

Wissam Siblini*,**, Pascale Kuntz**, Frank Meyer*

*Orange Labs
2 av. Pierre Marzin - 22 300 Lannion, France
wissam.siblini@univ-nantes.fr
**Laboratoire d'Informatique de Nantes Atlantique
Site Polytech Nantes – 44 300 Nantes cedex, France

Résumé. Des besoins actuels orientent la recherche en apprentissage multi-label interactif vers l'intégration d'un très grand nombre de variables à la fois en entrée et en sortie. Pour s'adapter à ce cadre, nous nous intéressons particulièrement à des algorithmes qui apprennent par l'intermédiaire d'une réduction de dimension. Dans cette étude, nous en comparons expérimentalement trois (SSI, CCA et Gravity) sur des données réelles et des données synthétiques engendrées pour tester leur capacité à extraire les informations pertinentes, leur tenue de charge et leur prise en compte d'un contexte très simple.

1. Introduction

Au cours de la dernière décennie, l'apprentissage multi-label a reçu une attention d'importance croissante, avec des applications variées (e.g. Zhang et al. 2015) telles que l'annotation d'images ou la catégorisation de documents. Son but est de prédire un vecteur binaire $\hat{y}$ dans un espace de labels $\mathcal{L} = \{0,1\}^{n_y}$ de dimension n_y à partir d'un vecteur (ici binaire) x défini sur un espace d'attributs $\mathcal{F}$ (ici $= \{0,1\}^{n_x}$) de dimension n_x. De nombreuses expérimentations ont été réalisées ces dernières années et ont fait émerger différentes familles d'algorithmes multi-label performants en prédiction (e.g. Madjarov et al., 2012). Récemment, des efforts se sont également concentrés sur la vitesse d'apprentissage et de prédiction de ces algorithmes afin de favoriser des applications interactives (Nair-Benrekia, et al., 2015). Des besoins applicatifs actuels (e.g. associations d'images et de textes, recommandations automatiques multicritères) orientent la recherche en apprentissage multi-label interactif vers l'intégration d'un très grand nombre de variables ($>10^5$) à la fois en entrée et en sortie.

Les algorithmes qui étaient expérimentalement les plus performants dans la littérature pour des dimensions restreintes ne sont plus adaptés à cause d'une trop forte complexité algorithmique et de la difficulté d'apprendre avec beaucoup de variables. Au contraire, les algorithmes qui apprennent par l'intermédiaire d'une réduction de

dimension et qui ont également fait leurs preuves sur les jeux de données multi-label (Luo, et al., 2014) devraient permettre un meilleur passage à l'échelle. Dans cette communication, nous nous intéressons alors à trois algorithmes de cette catégorie : Supervised Semantic Indexing (Bai, et al., 2009) et Canonical Correlation Analysis (Hotteling, 1936 ; Hardoon et al., 2004) qui effectuent une réduction de dimension supervisée par projections conjointes des espaces d'entrée et de sortie, et Gravity (Takacs, et al., 2007) basé sur une approche non supervisée utilisant un algorithme de factorisation de matrice très rapide.

Nous proposons ici une comparaison préliminaire des mesures de performances des algorithmes sur des données synthétiques engendrées pour contrôler les différents critères suivants : (i) la capacité à extraire l'information pertinente dans l'ensemble des attributs $\mathcal{F}$, (ii) la tenue de charge et la rapidité face à des jeux de données en très grande dimension et (iii) la capacité à prendre compte un contexte simple. L'analyse est complétée par les résultats sur un jeu de données réelles.

2. Principes généraux des algorithmes évalués

Supervised Semantic Indexing (SSI). SSI est un algorithme d'apprentissage supervisé de deux projections $\phi_{\mathcal{F}}$ de $\mathcal{F}$ et $\phi_{\mathcal{L}}$ de $\mathcal{L}$ où l'adéquation entre les éléments des espaces projetés est évaluée par un produit scalaire. Il a initialement été développé pour apprendre à évaluer et à ordonner la qualité de l'adéquation entre des requêtes textuelles $x \in \mathcal{F}$ et des documents $y \in \mathcal{L}$. Ici, la capacité à ordonner est utilisée pour prédire par sélection. Plus précisément, considérons un ensemble de triplets $\{(x^{(k)}, y^{(k)+}, y^{(k)-}) \in \mathcal{F} \times \mathcal{L} \times \mathcal{L}, k = 1, N_{\mathcal{T}}\}$ où $y^{(k)+} \in \mathcal{L}$ et $y^{(k)-} \in \mathcal{L}$ sont deux sorties respectivement en adéquation et inadéquation avec $x^{(k)} \in \mathcal{F}$ et $N_{\mathcal{T}}$ est la taille de l'espace d'apprentissage $\mathcal{T}$. SSI cherche les transformations linéaires $\phi_{\mathcal{F}}$ (de matrice $\mathrm{M}_{\mathcal{F}} \in \mathbb{R}^{m \times n_x}$) et $\phi_{\mathcal{L}}$ (de matrice $\mathrm{M}_{\mathcal{L}} \in \mathbb{R}^{m \times n_y}$) qui projettent cet ensemble de sorte que $\langle \phi_{\mathcal{F}}(x^{(k)}) | \phi_{\mathcal{L}}(y^{(k)+}) \rangle > \langle \phi_{\mathcal{F}}(x^{(k)}) | \phi_{\mathcal{L}}(y^{(k)-}) \rangle$. La résolution est effectuée par une descente de gradient qui vise à minimiser la fonction de perte suivante : $\sum_{k=1}^{N_{\mathcal{T}}} \max(0; 1 - \langle \Phi_{\mathcal{F}}(x^{(k)}) | \Phi_{\mathcal{L}}(y^{(k)+}) \rangle + \langle \Phi_{\mathcal{F}}(x^{(k)}) | \Phi_{\mathcal{L}}(y^{(k)-}) \rangle)$. Au cours de la minimisation, à chaque sélection d'un exemple $(x^{(k)}, y^{(k)})$ de $\mathcal{T}$, l'algorithme construit un triplet $(x^{(k)}, y^{(k)}, y^{(k')})$ où $y^{(k')}$ est tiré aléatoirement parmi les autres exemples. En prédiction, pour tout x de l'ensemble de test $\mathcal{S}$ (de taille N_S), la sortie proposée est $\hat{y}$ de $\mathcal{T}$ qui maximise la fonction de score $\langle \Phi_{\mathcal{F}}(x) | \Phi_{\mathcal{L}}(y) \rangle$.

Corrélation Canonique (CCA). CCA cherche à déterminer les projections $\mathcal{F}'$ de $\mathcal{F}$ et $\mathcal{L}'$ de $\mathcal{L}$ ayant les corrélations maximales. Il s'agit donc de trouver des couples $(w_x, w_y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ tels que la corrélation canonique ρ entre $\langle w_x | x \rangle$ et $\langle w_y | y \rangle$ soit maximale sur l'ensemble d'apprentissage $\mathcal{T}$. Ce problème se ramène à un pro-

blème aux valeurs propres généralisées de type $Aw = \lambda Bw$ et les m meilleures solutions (w_x, w_y) sont liées aux vecteurs propres associés aux m plus grandes valeurs propres de (A, B). En prédiction, pour tout x de $\mathcal{S}$, on recherche la sortie $\hat{y}$ dans $\mathcal{T}$ qui minimise la distance euclidienne $\|\mathrm{M}_{\mathcal{F}}.x - \mathrm{M}_{\mathcal{L}}.y\|_2$ où $\mathrm{M}_{\mathcal{F}} \in \mathbb{R}^{m \times n_x}$ et $\mathrm{M}_{\mathcal{L}} \in \mathbb{R}^{m \times n_y}$ sont les matrices constituées des m vecteurs w_x et w_y déterminés pendant l'apprentissage.

Gravity. Il s'agit d'un algorithme de factorisation rapide de matrice initialement développé pour la recommandation automatique par filtrage collaboratif. Il est utilisé ici pour factoriser les entrées de test et d'apprentissage afin d'en réduire le nombre d'attributs avant de prédire par recherche du plus proche voisin. Considérons la matrice $M_{int} \in \{0,1\}^{(N_{\mathcal{T}}+N_{\mathcal{S}}) \times n_x}$ constituée de l'ensemble des entrées disponibles x de $\mathcal{T}$ et de $\mathcal{S}$. Gravity factorise la représentation creuse M de M_{int} où seules les valeurs non nulles sont encodées : il détermine deux matrices $P \in \mathbb{R}^{m \times (N_{\mathcal{T}}+N_{\mathcal{S}})}$ et $Q \in \mathbb{R}^{m \times n_x}$ de sorte que M coïncide au mieux avec la matrice pleine $P^T Q$ sur ses composantes non nulles. Les matrices P et Q sont déterminées par une descente de gradient stochastique avec régularisation. Chaque colonne de P est ensuite considérée comme la version factorisée de chaque entrée x de $\mathcal{T}$ et de $\mathcal{S}$. Pour tout x de $\mathcal{S}$, la sortie $\hat{y}$ prédite est la sortie associée à son plus proche voisin x_v dans $\mathcal{T}$.

3. Expérimentations

Pour tester le comportement des algorithmes sur les trois critères retenus nous avons construit trois de jeux de données adaptés à chaque critère avec lesquels nous avons effectué 10 tests (10-fold).

Jeu de données $\mathcal{D}$	n_x	d_x	n_y	d_y	N_{tot}
Extraction de pertinence $\mathcal{D}_p$	2000	30	1000	9.5 (4.4)	200000
Tenue de charge $\mathcal{D}_t$	10^5	30	10^4	30	10^6
Gestion du contexte $\mathcal{D}_c$	2001	30	2000	14.8 (2.7)	200000
Ohsumed BOW $\mathcal{D}_r$	329769	117 (95)	14627	11(4)	348566

TAB. 1 – *Caractéristiques des jeux de données synthétiques ($\mathcal{D}_p$,$\mathcal{D}_t$,$\mathcal{D}_c$) et réel $\mathcal{D}_r$. Les valeurs indiquées sont la taille n_x de l'espace d'attributs en entrée, la taille n_y de l'espace de labels en sortie, les nombres moyens –et écarts-types- d'attributs et de labels non nuls d_x et d_y respectivement en entrée et en sortie, et N_{tot} le nombre total d'exemples.*

Jeu de données $\mathcal{D}_p$ (capacité à extraire l'information pertinente dans l'espace des attributs). Chaque attribut x_i de x est généré aléatoirement à 0 ou 1 et chacun des labels y_i est fonction de la somme d'un sous-ensemble de taille s (ici s = 30) de x_i

choisis parmi les n_p premiers attributs de x (ici $n_p = n_x/2$). La binarisation des valeurs de y_i est arbitrairement définie comme suit : $y_i = 1$ si la somme des s valeurs de x_i sélectionnées est supérieure ou égale à 2 ; sinon $y_i = 0$. L'objectif est d'évaluer la capacité de l'algorithme à s'appuyer uniquement sur les n_p premiers attributs « informatifs » de x.

Jeu de données $\mathcal{D}_t$ (tenue de charge). La génération de x est identique à celle de $\mathcal{D}_p$ et chaque label y_i de y est fonction de 10 attributs x_i sélectionnés par un tirage aléatoire sans remise : $y_i = 1$ si au moins un des x_i sélectionné vaut 1 et $y_i = 0$ sinon. On engendre ainsi un million d'exemples avec $n_x = 100000$ et $n_y = 10000$.

Jeu de données $\mathcal{D}_c$ (capacité à prendre en compte un contexte simple). La génération de x est identique à celle de $\mathcal{D}_p$ à l'exception du premier attribut x_c (contexte) qui vaut 1 ou 0 avec une équiprobabilité de 0.5. Notons x' le vecteur des r (ici $r = 1000$) attributs suivants de x et x'' le vecteur des $n_x - r - 1$ suivants. Les labels de y sont définis comme suit : $y_i = x'_i$ si $x_c = 1$ ou $y_{i+r} = x''_i$ si $x_c = 0$.

Jeu de données réel Ohsumed (Hersh, et al., 1994). Il est constitué d'informations concernant 348566 publications médicales. L'objectif est prédire à partir du titre et du résumé (traités en sac de mots) les mots-clés de ces publications. Aucun traitement de langage n'est appliqué, le vocabulaire des textes en entrée est en dimension $n_x \approx 330000$ et le vocabulaire des mots-clés est en dimension $n_y \approx 15000$.

Les performances des algorithmes sont évaluées selon quatre mesures classiques qui sont pertinentes dans notre contexte : la précision, le rappel, le F_1-score et la complexité algorithmique.

4. Résultats et perspectives

Les algorithmes sont comparés à une « baseline » qui compte l'occurrence de 1 de chaque label y_i dans les exemples d'apprentissage. Elle prédit la même sortie $\hat{y}$, quel que soit x, dans laquelle seuls les d_y labels les plus occurrents valent 1. De plus, comme cette étude préliminaire se focalise sur l'analyse des comportements relatifs des algorithmes, nous les avons tous restreints à la recherche d'espaces réduits de même dimension ($m = 100$). Pour la prédiction, nous les avons limités à la recherche du meilleur candidat parmi les sorties de l'apprentissage. Une sortie ne peut donc être prédite que par une autre sortie y du jeu de donnée et cela restreint le F_1-score maximal accessible par les algorithmes. Nous avons évalué ce maximum pour chaque jeu de données. Les résultats sont présentés dans le Tableau 2 ci-dessous.

En réduisant conjointement les dimensions de $\mathcal{F}$ et de $\mathcal{L}$, CCA et SSI sont tous les deux capables d'extraire uniquement les attributs pertinents de $\mathcal{F}$ pour présenter les meilleures performances sur $\mathcal{D}_p$. En revanche, les entrées et sorties de $\mathcal{D}_t$ sont difficiles à réduire car elles sont formées de nombreux attributs et labels pertinents.

Elles sont plus faciles à résumer sur les données réelles $\mathcal{D}_r$ malgré des dimensions comparables. Notons également que la linéarité des algorithmes est un frein à la gestion du contexte dans $\mathcal{D}_c$. Néanmoins SSI et CCA capturent en partie des liens entre les plages d'attributs x', x'' et de labels y.

	$\mathcal{D}_p$			$\mathcal{D}_t$			$\mathcal{D}_c$			$\mathcal{D}_r$		
	P	R	F	P	R	F	P	R	F	P	R	F
CCA	**24.67** (**0.44**)	11.46 (0.18)	**14.69** (**0.25**)				3.92 (0.15)	3.51 (0.11)	3.67 (0.12)			
Gravity	2.36 (0.10)	2.58 (0.40)	2.45 (0.32)	0.25 (0.01)	0.28 (0.04)	0.25 (0.01)	0.72 (0.06)	0.73 (0.08)	0.71 (0.05)	**31.05** (**0.30**)	25.10 (0.30)	25.99 (0.15)
SSI	6.47 (0.05)	**14.17** (**0.09**)	8.37 (0.05)	**0.60** (**0.00**)	**0.90** (**0.00**)	**0.72** (**0.00**)	**5.23** (**0.17**)	**6.66** (**0.22**)	**5.77** (**0.18**)	17.80 (0.30)	**32.70** (**0.73**)	21.41 (0.53)
Baseline	2.18 (0.03)	2.42 (0.04)	2.27 (0.03)	0.30 (0.00)	0.30 (0.00)	0.30 (0.00)	0.75 (0.04)	0.55 (0.06)	0.63 (0.03)	27.95 (0.04)	28.75 (0.04)	**28.23** (**0.04**)
Maximum			44.49			12.72			28.34			67.76

TAB.2 – *Résultats pour les jeux de données synthétiques et le jeu de données réel pour les trois algorithmes. Les valeurs moyennes (en %) et les écart-types des 10 tests sont indiqués. P = Précision, R = Rappel, F = F1-score. Le maximum accessible est indiqué en dernière ligne.*

Gravity ne fournit pas de bons résultats sur des jeux synthétiques. Les attributs en entrée étant non correlés par construction, la factorisation non supervisée qu'il effectue induit une forte perte d'information pertinente. Mais les résultats sont meilleurs sur $\mathcal{D}_r$ car les textes représentés dans $\mathcal{F}$ en entrées sont corrélés entre eux : la représentation latente créée compresse l'information utile pour prédire y. Sur le jeu $\mathcal{D}_r$, SSI parvient également à apprendre. Et la baseline est performante car les $d_y = 11$ mots-clés les plus occurrents sont présents dans de nombreux exemples.

Sans utilisation de méthodes de recherche optimale, la complexité des algorithmes en prédiction est la même et de l'ordre de $N_S \times N_\mathcal{T} \times (d_x + d_y) \times m$. En apprentissage, Gravity ($O(n_{it} \times (N_\mathcal{T} + N_S) \times d_x \times m)$) et SSI ($O(n_{it} \times N_\mathcal{T} \times (d_x + d_y) \times m)$ sont nettement moins complexes que CCA ($O(n_x^3 + n_y^3 + N_\mathcal{T} \times (d_x + d_y))$) et sont donc susceptibles de mieux tenir la charge. De plus, les complexités temporelles (>> 24h) et spatiales (>16Go de RAM) de CCA sont trop importantes pour fournir des résultats sur $\mathcal{D}_t$ et $\mathcal{D}_r$ avec un calculateur « standard ». (Processeur Intel® Core™ i5-2520M 2.5 GHz, 16Go de RAM)

Différentes voies d'amélioration émergent de cette première analyse. En apprentissage, la réduction conjointe de $\mathcal{F}$ et $\mathcal{L}$ est prometteuse mais doit permettre le traitement des données creuses et assurer la tenue de charge. Trois pistes s'ouvrent dans cette direction : la supervision du très rapide Gravity, une recherche de valeurs propres moins coûteuse pour CCA, et l'amélioration du compromis vitesse/précision

de SSI. De plus, il faudra évaluer des versions non linéaires de ces algorithmes pour pouvoir gérer en grande dimension des interactions complexes entre attributs. Il sera enfin nécessaire d'élargir le domaine de prédiction notamment par la recherche rapide et la combinaison optimale de $p > 1$ meilleurs candidats.

Références

Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., & Weinberger, K. (2009). *Supervised semantic indexing*. Proc. of the 18th ACM Conf. on Information and Knowledge Management, pp. 187-196.

Luo, G., Huang, T., & Shi, Z. (2014). *Multi-label Classification Using Hypergraph Orthonormalized Partial Least Squares*. Journal of Computers, 9(6), 1364-1370.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). *Canonical correlation analysis: An overview with application to learning methods*. Neural computation, 16(12), 2639-2664.

Hotelling, H. (1933). *Analysis of a complex of statistical variables into principal components*. Journal of educational psychology, 24(6), 417.

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). *An extensive experimental comparison of methods for multi-label learning*. Pattern Recognition, 45(9), 3084-3104.

Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994). *OHSUMED: An interactive retrieval evaluation and new large test collection for research*. In SIGIR'94 (pp. 192-201). Springer London.

Nair-Benrekia, N. Y., Kuntz, P., & Meyer, F. (2015). *Learning from multi-label data with interactivity constraints: an extensive experimental study*. Expert Systems with Applications, 42(13), 5723-5736.

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2007).*Major components of the gravity recommendation system*. ACM SIGKDD Explorations Newsletter, 9(2), 80-83.

Zhang, M. L., & Wu, L. (2015). *LIFT: Multi-label learning with label-specific features*. IEEE Trans. on Pattern Analysis and Machine Intelligence, 37(1), 107-120.

Summary

Research in interactive multi-label learning is currently driven toward the management of a large number of variables in both inputs and outputs. To efficiently perform in this context, we focus on algorithms that learn via dimension reduction. In this paper, we experimentally compare three candidate algorithms (SSI, CCA and Gravity) on a real dataset and on synthetic datasets especially designed to test their ability to extract relevant information, to perform on high-dimensional data and to manage a very simple context.