Sélection d'une méthode de classification multi-label pour un système interactif

Noureddine Yacine Nair Benrekia[*,**], Pascale Kuntz[**], Franck Meyer[*]

[*]Orange Labs, Av. Pierre Marzin 22307 Lannion cedex France
(yacinenoureddine.nairbenrekia, franck.meyer)@orange.com,
[**]LINA, la Chantrerie-BP 50609, 44360 Nantes cedex France
pascale.kuntz@univ-nantes.fr

Résumé. L'objectif de cet article est d'évaluer la capacité de 12 algorithmes de classification multi-label à apprendre, en peu de temps, avec peu d'exemples d'apprentissage. Les résultats expérimentaux montrent des différences importantes entre les méthodes analysées, pour les 3 mesures d'évaluation choisies: Log-Loss, Ranking-Loss et Temps d'apprentissage/prédiction, et les meilleurs résultats sont obtenus avec: multi-label k Nearest neighbours (ML-kNN), suivi de Ensemble de Classifier Chains (ECC) et Ensemble de Binary Relevance (EBR).

1 Introduction

Les systèmes de classification automatique usuels ne permettent pas d'interagir directement avec un algorithme d'apprentissage, et par conséquent, les résultats qu'ils produisent sont, en pratique, souvent en décalage avec les points de vue des utilisateurs. Pour personnaliser ces systèmes, une solution est d'intégrer l'utilisateur, dans le processus d'apprentissage, pour qu'il génère, explicitement via un support visuel, ses propres classifieurs (Ware et al., 2001). L'utilisateur devient donc le coach qui va annoter, en positif ou négatif, un nombre limité d'exemples pour entraîner un algorithme à apprendre ses préférences. Ensuite, cet algorithme doit être capable de généraliser et de produire des prédictions personnalisées pour le reste des exemples non-classés. Interactivement, l'utilisateur peut corriger les mauvaises prédictions afin de renforcer le modèle. De tels systèmes d'apprentissage ont récemment suscité un intérêt croissant et ont trouvé des applications dans plusieurs domaines (e.g *CueFLIK* pour la classification d'images et *CueT* pour la classification d'alarmes (Amershi, 2011)).

Dans le domaine télévisuel, il existe des systèmes de recommandation automatique (Bambini et al., 2011) mais ils sont, à notre connaissance, mono-label et ne prennent pas en compte des labels subjectifs de perception qualitative (e.g. « ce film me fait peur ») qui sont à l'évidence importants pour les utilisateurs. Dans ce contexte, notre objectif final, qui dépasse le cadre de cet article, est de concevoir un système de classification interactive personnalisée pour mieux assister les utilisateurs dans leur recherche de contenus numériques. Ce travail requiert en préalable un état des lieux sur les capacités respectives des techniques de classification multi-label adaptées aux contraintes d'interactivité. Nous nous focalisons donc ici

sur une comparaison expérimentale de 12 méthodes de classification multi-label avec des mesures d'évaluation adaptées à notre cas d'utilisation final, et des jeux de données classiques de difficulté croissante. En complément, nous effectuons une première évaluation des temps d'apprentissage et de prédiction, ainsi que de la vitesse de généralisation des classifieurs en variant le nombre d'exemples d'apprentissage.

La suite de cet article est organisée de la manière suivante : la section 2 définit plus précisément les objectifs de notre étude. La section 3 présente les 12 méthodes de classification multi-label sélectionnées pour cette comparaison. Les jeux de données d'évaluation et le protocole expérimental sont décrits dans la section 4. Les résultats obtenus sont résumés et discutés dans la section 5.

2 Définition du problème

Soit $\mathcal{F} = \{f_1, \ldots, f_j, \ldots, f_m\}$ un espace de m attributs tel que $dom(f_j) \in \mathcal{R}$, et $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k, \ldots, \lambda_q\}$ un espace de q labels tel que $dom(\lambda_k) \in \{0, 1\}$ (0 : non-pertinent, 1 : pertinent). Soit $\mathcal{D} = \{(x_i, y_i)|\ i = 1..|\mathcal{D}|\}$ un jeu de données multi-label. Chaque exemple $x_i = \langle x_i^1, \ldots, x_i^j, \ldots, x_i^m \rangle$ est annoté par un ensemble de labels $y_i = \langle y_i^1, \ldots, y_i^k, \ldots, y_i^q \rangle$, avec $dom(x_i) \in \mathcal{R}^m$, $dom(y_i) \in \{0, 1\}^q$ et $|y_i| \leq q$ où $|y_i|$ et $|\overline{y_i}|$ représentent respectivement le nombre de labels pertinents et non-pertinents de x_i. Un classifieur multi-label h a pour objectif de prédire les labels $\hat{y_i}$ des exemples non-étiquetés $x_i \in \mathcal{S}$ à partir d'un ensemble d'apprentissage $\mathcal{T} \subset \mathcal{D}$ de taille limitée, où $\mathcal{S}$ est un jeu de test de taille relativement grande comparée à celle de $\mathcal{T}$, i.e. $|\mathcal{T}| << |\mathcal{S}|$. Plus formellement, $\hat{y_i} = h(x_i) = \langle \hat{y_i^1}, \ldots, \hat{y_i^k}, \ldots, \hat{y_i^q} \rangle$ avec $dom(\hat{y_i}) \in [0..1]^q$.

Récemment, (Madjarov et al., 2012) ont proposé une comparaison expérimentale étendue des méthodes d'apprentissage multi-label et ont recommandé RF-PCT, HOMER, BR et CC en tant que classifieurs de référence. Cependant, nous ajoutons au problème d'apprentissage multi-label les contraintes d'interactivité. Ainsi, un « bon » classifieur doit être en mesure de fournir de « bonnes » prédictions, en un temps court, avec peu d'information supervisée.

Pour qu'un classifieur soit approprié à notre futur système de classification de *VoD*, il doit vérifier 3 propriétés essentielles : (**1**) pour chaque exemple sélectionné par l'utilisateur, les premiers labels qu'il suggère doivent être plus discriminants et plus pertinents que les labels suivants ; (**2**) les prédictions de labels qu'il fournit à l'utilisateur pour un exemple sélectionné doivent être proches des vrais labels, et (**3**) pour apprendre un modèle et fournir des prédictions à l'utilisateur, il doit être le plus rapide possible. L'évaluation de ces propriétés est basée sur les trois mesures suivantes.

La mesure Rank-Loss (Madjarov et al., 2012) permet d'évaluer la conservation de l'ordre des labels :

$$Ranking - Loss \downarrow = \frac{1}{v} \sum_{i=1}^{v} \frac{1}{|y_i| \times |\overline{y_i}|} |(\lambda_a, \lambda_b) \in y_i \times \overline{y_i} : r_i(\hat{y_i^a}) < r_i(\hat{y_i^b})|$$

La mesure Log-Loss (Read, 2010) permet d'évaluer le taux de dissimilarité entre les vrais labels et les labels prédits :

$$Log - Loss \downarrow = \frac{1}{v \times q} \sum_{i=1}^{v} \sum_{j=1}^{q} min(-(ln(\hat{y_i^j}) \times y_i^j + ln(1 - \hat{y_i^j}) \times (1 - y_i^j)), ln(v))$$

L'efficacité de résolution des algorithmes est évaluée ici par les temps d'apprentissage et de prédiction en terme de secondes.

3 Méthodes de classification multi-label

Les approches de classification multi-label se divisent en trois grandes familles selon (Madjarov et al., 2012) : **1) Méthodes de transformation** : elles transforment le problème d'apprentissage multi-label en plusieurs problèmes de classification ou régression mono-label, **2) Méthodes adaptées** : elles customisent des algorithmes d'apprentissage mono-label pour les adapter au cas multi-label et **3) Méthodes ensemble** : elles utilisent des ensembles de classifieurs issus de la première ou la deuxième famille d'approches.

Pour notre comparaison expérimentale, nous évaluons 12 classifieurs multi-label, parmi lesquels nous retrouvons tous les classifieurs recommandés par (Madjarov et al., 2012) sauf RF-PCT que nous étudierons dans une prochaine étude expérimentale. Le choix des méthodes est fonction de leur fréquence d'utilisation dans la littérature, la disponibilité de leur implémentation et de leur appartenance aux différentes classes de méthodes. Les implémentations de ces algorithmes sont disponibles sur MEKA [1]. Dans tous les cas, les classifieurs sont exécutés avec leurs paramètres par défaut à l'exception de ML-kNN pour lequel, étant donné la petite taille des ensembles d'apprentissage, nous avons fixé son paramètre k à 1.

Dans la $1^{ère}$ famille, nous sélectionnons 5 méthodes : **(1)** Binary Relevance (**BR**), **(2)** Classifier Chain (**CC**), **(3)** Label Powerset (**LP**), **(4)** Calibrated Label Ranking (**CLR**) (Madjarov et al., 2012), et **(5)** une méthode qui retourne toujours la combinaison de labels la plus fréquente dans l'ensemble d'apprentissage (**Baseline**). Les méthodes de transformation utilisent des classifieurs binaires et les plus utilisés sont : Support Vector Machine (SVM) et l'arbre de décision C4.5 (Read, 2010; Madjarov et al., 2012). Nous avons choisi ici C4.5 comme classifieur de base pour de meilleures performances en temps d'apprentissage car il exploite un sous-ensemble d'attributs contrairement à SVM qui requiert la totalité de l'ensemble. Ce choix est important dans notre contexte de *VoD* où le nombre d'attributs peut être très important.

Dans la $2^{ème}$ famille, nous choisissons 2 méthodes : **(1) AdaBoost.MH** et **(2) ML-kNN** (Madjarov et al., 2012). Enfin, dans la $3^{ème}$ famille, nous évaluons 3 méthodes : **(1)** RAndom k labEL sets (**RAkEL**), **(2)** Hierarchy Of multi-label classifiERs (**HOMER**), **(3)** Ensemble de Classifier Chains (**ECC**) (Madjarov et al., 2012) et Ensemble de Binary Relevance (**EBR**) (Read, 2010).

4 Evaluation

Pour la comparaison expérimentale des classifieurs, nous avons sélectionné 5 jeux de données souvent utilisés dans la littérature multi-label (Tab. 1). La dimensionnalité de leurs espaces d'attributs est petite comparée à la grande dimensionnalité d'un catalogue de *VoD*. Néanmoins, elles fournissent un premier aperçu sur le comportement de chaque classifieur. Afin d'évaluer la performance prédictive et la rapidité de chacun des classifieurs, nous avons élaboré un protocole expérimental simple qui simule la phase initiale dans laquelle l'utilisateur crée

1. Une libraire multi-label disponible sur : http ://meka.sourceforge.net/

TAB. 1 – *Description des jeux données avec DL : nombre de combinaisons de labels différentes et $Lcard$: nombre de labels associé en moyenne aux exemples.*

Corpus	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{L}\rvert$	DL	$Lcard$
Emotions	72	592	118	474	6	27	1.87
Yeast	103	2417	483	1934	14	198	4.24
Scene	294	2407	481	1926	6	15	1.07
Slashdot	1079	3782	756	3026	22	156	1.18
Imdb	1001	7500	1500	6000	28	1021	2.00

progressivement son ensemble d'apprentissage et attend en retour des premières prédictions satisfaisantes.

Plus précisément, nous divisons chaque corpus $\mathcal{D}$ en 5 parties disjointes pour effectuer une 5-validation croisée : une partie $\mathcal{A}$ constitue l'ensemble d'apprentissage et les 4 autres parties constituent l'ensemble de test $\mathcal{S}$. À partir de $\mathcal{A}$, nous extrayons w ensembles de p sous-ensembles u_i imbriqués. Le $1^{èr}$ sous-ensemble u_1 est de taille $|u_1| = 2^1$, le $2^{ème}$ est de taille $|u_2| = 2^2$, ..., et le $p^{ème}$ est de taille $|u_p| = 2^p$. Ensuite, pour chaque mesure, nous évaluons la performance moyenne de chaque classifieur sur les 5 bases de test de chaque corpus : chaque classifieur est entraîné avec tous les sous-ensembles d'apprentissage de u_1 à u_p. Puis, pour chaque classifieur, nous calculons, pour chaque taille de sous-ensemble d'apprentissage, la performance moyenne sur les 5 corpus.

Dans toutes nos expérimentations, w a été fixé à 20 (i.e., 100 ensembles testés pour les 5 validations croisées) et p à 6. Le nombre d'ensembles testés est suffisant pour obtenir une performance moyenne de chaque classifieur avec un faible écart-type, et la taille maximale d'un sous-ensemble d'apprentissage (i.e., 64) est conforme au nombre d'exemples qu'un utilisateur pourrait annoter au maximum dans un cas d'usage réel.

5 Résultat expérimentaux

Les résultats présentés, dans les graphiques de la Fig. 1, sont des moyennes et des écarts-types de performances sur les 5 jeux de données. Les classifieurs sont ordonnés en fonction de leur performance globale qui tient compte de leurs performances moyennes obtenues pour toutes les tailles des sous-ensembles d'apprentissage.

5.1 Log-Loss pour la prédiction des labels

Comme nous le constatons sur la Fig. 1, ML-kNN et AdaBoost.MH surclassent tous les classifieurs avec un léger avantage pour ML-kNN. En outre, les plus mauvais résultats sont obtenus par LP, BR, CC et notre Baseline. Par ailleurs, les 7 meilleurs algorithmes et LP sont les seuls à pouvoir améliorer leurs performances prédictives quand le nombre d'exemples d'apprentissage double.

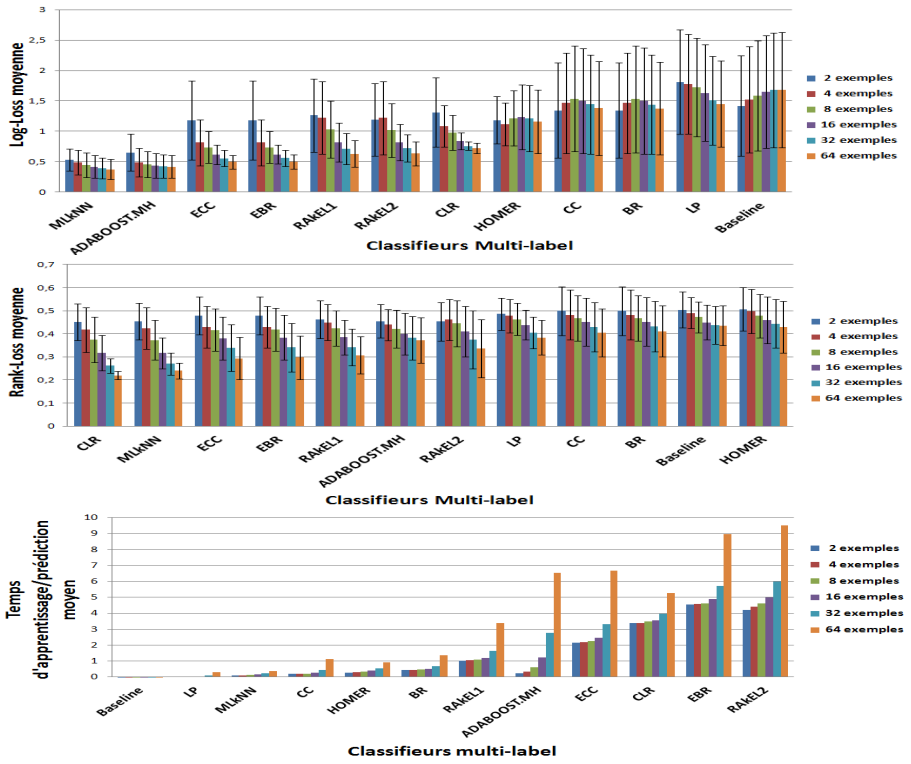

FIG. 1 – *Performances moyennes des classifieurs multi-label en Log-Loss, Rank-Loss, temps d'apprentissage/prédiction (de haut en bas).*

5.2 Rank-Loss pour l'ordonnancement des labels

Globalement, les classements des classifieurs avec les mesures Log-Loss et Rank-Loss sont assez semblables car elles sont intuitivement corrélées. Cependant, CLR devient la meilleure approche pour Rank-Loss et Adaboost.MH passe de la $2^{ème}$ place à la $6^{ème}$ place. Il n'est pas surprenant que CLR obtienne les meilleurs résultats car elle a été conçue spécialement pour améliorer la qualité du classement des labels. De même, ML-kNN vise à minimiser cette mesure. En outre, lorsque le nombre d'exemples d'apprentissage augmente, CLR et ML-kNN sont les plus efficaces pour améliorer la qualité du classement des labels.

5.3 Temps d'apprentissage/prédiction

Comme toutes nos expérimentations ont été menées avec des implémentations de la librairie MEKA, que nous ne maîtrisons pas, les cumuls des temps d'apprentissage et de prédictions calculés ne donnent qu'une tendance de la complexité algorithmique (Fig. 1). Par exemple, EBR devrait être plus rapide que ECC mais nous observons ici le contraire. Parmi les meilleures approches (ECC, EBR et ML-kNN) pour les mesures précédentes, ML-kNN

semble le plus rapide. En effet, il n'apprend pas de modèle mais nécessite seulement quelques millisecondes pour estimer les probabilités a priori/a posteriori à partir du sous-ensemble d'apprentissage, et moins d'une demi seconde pour la prédiction parce que le nombre de voisins fixé est faible ici (i.e., k = 1). Par ailleurs, EBR et ECC nécessitent plus de temps mais sont suffisamment rapides pour terminer dans les premières secondes.

6 Conclusion et Travaux futurs

Nous avons proposé, dans cet article, une comparaison expérimentale de 12 classifieurs differents avec des contraintes d'interactivité fortes qui se traduisent par des exigences en terme de temps d'apprentissage et de prédiction d'une part, et de vitesse de convergence en nombre d'exemples d'apprentissage d'autre part. Nous avons évalué ces algorithmes sur 5 jeux de données de petites tailles avec 4 mesures adaptées à notre cas d'utilisation. Notre protocole a montré que les meilleures performances prédictives ont été obtenues par ML-kNN, suivi de ECC et EBR.

Dans une prochaine étape, nous allons étendre cette étude expérimentale avec d'autres approches intéressantes de classification multi-label telle que RF-PCT, et nous sommes en train d'analyser le passage à l'échelle des différentes approches afin de s'approcher des tailles de données réelles rencontrées dans le domaine de la *VoD*.

Références

Amershi, S. (2011). Designing for effective end-user interaction with machine learning. In *Proc. of the 24th annual ACM symposium adjunct on User interface software and technology*, pp. 47–50. ACM.

Bambini, R., P. Cremonesi, et R. Turrin (2011). A recommender system for an IPTV service provider : a real large-scale production environment. In *Recommender Systems Handbook*, pp. 299–331. Boston, MA : Springer US.

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

Read, J. (2010). *Scalable Multi-label Classification*. Ph. D. thesis, University of Waikato.

Ware, M., E. Frank, G. Holmes, M. Hall, et I. H. Witten (2001). Interactive machine learning : letting users build classifiers. *Int. J. of Human-Computer Studies 55*(3), 281–292.

Summary

The objective of this paper is to evaluate the ability of 12 multi-label classification algorithms at learning, in a short time, with few training examples. Experimental results highlight significant differences for 3 selected evaluation measures: Log-Loss, Ranking-Loss, Learning/Prediction time, and the best results are obtained with: Multi-label k Nearest neighbors (ML-kNN), followed by Ensemble of Classifier Chains (ECC) and Ensemble of Binary Relevance (EBR).