LASCAR : LArge Scale CAtegoRization

Classification dans les très grands systèmes de catégories

A. Antionadis, M. Burlet, Y. Denneulin, E. Gaussier

Laboratoire Jean Kuntzmann (LJK)

Laboratoire des sciences pour la conception, l'optimisation et la production (G-SCOP)

Laboratoire d'Informatique de Grenoble (LIG)

1 Introduction

Plusieurs problèmes de catégorisation mettent en jeu des systèmes comprenant plusieurs milliers de catégories. Les offices de brevets, par exemple, sont chargés d'affecter, à chaque nouvelle demande de brevet, un code fondé sur la Classification Internationale des Brevets[1] qui contient environ 70 000 sous-divisions. DMOZ[2], qui se veut le plus grand répertoire du web, contient plus de 590 000 catégories, dans lesquelles de nouvelles pages sont catégorisées par une équipe de volontaires travaillant chacune sur une sous-partie du système complet. Une telle situation se rencontre aussi dans le cadre de l'annotation sémantique d'éléments, où il s'agit d'affecter à une partie d'un document un ou plusieurs concepts d'une ontologie ou d'un thésaurus. Dans le domaine médical par exemple, PubMed[3] contient plus de 16 millions de références dont les résumés sont indexés à partir des concepts du MeSH[4], un thésaurus qui contient plus de 150 000 concepts.

Dans tous ces cas, pour des raisons de maintenance et de navigation, les catégories (ou concepts) sont organisées de façon hiérarchique, généralement sous forme d'arbre (ou de forêt), parfois sous forme de graphe orienté sans cycle (c'est le cas de DMOZ ; cependant, peu de liens violent la structure d'arbre sous-jacente). La profondeur de tels arbres varie d'un domaine à un autre : MeSH comporte onze niveaux, alors que la CIB n'en a que quatre. De plus, toujours pour des raisons de maintenance et de navigation, le nombre de sous-catégories de chaque catégorie est relativement petit, de l'ordre de quelques dizaines. Enfin, seul un sous-ensemble des catégories (en général les feuilles) est utilisé pour archiver ou annoter un document, les catégories intermédiaires servant essentiellement à l'organisation du système de catégories. Nous qualifions de *finales* les catégories appartenant à ce sous-ensemble.

Le problème qui nous intéresse ici est celui de la catégorisation dans les grands systèmes de catégories. Il peut se formuler de la façon suivante :

Etant donné un grand ensemble de catégories, comment catégoriser de façon précise un nouveau document ?

Le défi qui se pose ici est bien sûr lié à la taille des données considérées : **grand nombre de catégories**, **grand nombre d'exemples**, **grand nombre d'attributs**.

Même si plusieurs travaux, menés dans des communautés de recherche différentes, ont abordé le problème du passage à l'échelle, il n'existe pas de classifieurs permettant de traiter un grand nombre de catégories ou de tenir compte de toutes les dépendances entre catégories. Dans la communauté base de données par exemple, [11] propose un classifieur parallèle fondé sur des arbres de décisions qui permet de traiter des exemples comportant un grand nombre d'attributs. De façon similaire, la technique de la pyramide, développée dans [2], permet de traiter des requêtes dans des espaces de grande dimension (c'est-à-dire dont les exemples comportent un grand nombre d'attributs). Dans la communauté apprentissage, [12] a récemment proposé une version semi-supervisée des séparateurs à vaste marge (SVMs) pouvant traiter un grand nombre d'exemples dans des espaces vectoriels de grande dimension. Des essais similaires ont été proposés dans [5, 7, 1] où la complexité de l'apprentissage d'un SVM est contournée par un

[1] CIB - http ://www.wipo.int/classifications/ipc/en/

[2] http ://dmoz.org/

[3] http ://www.ncbi.nlm.nih.gov/sites/entrez

[4] http ://www.nlm.nih.gov/pubs/factsheets/mesh.html

partitionnement de l'ensemble d'exemples, et par des procédures d'apprentissage et de mise à jour conduites en parallèle. Toutefois, dans tous ces cas, le nombre de catégories considéré est limité.

Certains travaux ont été menés pour tenir compte des dépendances entre catégories, telles que celles existant dans une hiérarchie de catégories. [6], par exemple, propose un classifieur probabiliste qui tient compte des dépendances entre catégories modélisées sous forme de graphe orienté sans cycle. Il en va de même pour les classifieurs fondés sur les champs aléatoires conditionnels (*conditional random fields*, [9, 10]). Cependant, ici encore, le passage à l'échelle ne peut se faire directement, et les algorithmes sous-jacents ne permettent pas de traiter un grand nombre de catégories. Un autre type d'approche permettant de rendre compte des dépendances entre catégories est fourni par le cadre développé dans [13, 14, 15], qui consiste à apprendre des séparateurs à vaste marge sur des sorties structurées et interdépendantes. En particulier, [14] montre que le nombre de contraintes utilisé dans la formulation duale du problème de maximisation de la marge ne dépend pas du nombre de catégories, mais plutôt du nombre d'exemples et d'une distance entre catégories. Ici encore, la limitation sur le nombre d'exemples ne permet pas d'utiliser directement ces approches sur les cas mentionnés plus haut.

Nous abordons directement dans ce projet le problème de la catégorisation sur un grand nombre de catégories, mettant en jeu un grand nombre d'exemples comportant un grand nombre d'attributs. Pour cela, nous retenons deux axes de recherche. Le premier, que nous qualifions de *top-down*, vise à déployer n'importe quelle technologie de catégorisation sur des grands systèmes de catégories en exploitant la structuration hiérarchique de ces systèmes. Le deuxième, que nous qualifions de *direct*, vise à définir une nouvelle technologie de catégorisation se concentrant directement sur l'ensemble des catégories finales.

Ce projet est à notre connaissance la première tentative de formalisation complète du problème de catégorisation sur les grands systèmes de catégories, et nous en attendons les retombées suivantes :

1. Un ensemble de résultats théoriques sur la catégorisation et le comportement des classifieurs dans des grands systèmes de catégories ;
2. Un ensemble de nouvelles méthodes et d'algorithmes associés exploitant ces résultats ;
3. Un ensemble d'outils permettant de catégoriser des documents dans de très grands systèmes de catégories.

Les résultats obtenus seront publiés dans les conférences et journaux importants du domaine. Les outils développés devraient être utiles aux organisations manipulant de grands systèmes de catégories, comme DMOZ, les offices nationaux et internationaux de brevets, ou Yahoo !. Nous conduirons des essais en grandeur réelle sur des collections associées, et construirons un démonstrateur pour présenter nos résultats à ces organisations. Enfin, d'autres domaines, comme celui de la bio-informatique, pourront bénéficier des résultats de nos recherches.

2 Programme de travail

2.1 Approche top-down

Il existe une approche simple permettent de déployer une technologie de catégorisation sur de grands systèmes de catégories tout en exploitant la structure de ce système. Cette approche, utilisée par exemple dans [16, 4], consiste à considérer tous les ensembles de catégories, niveau par niveau, et à apprendre un classifieur sur chacun de ces ensembles. Ainsi, l'ensemble des catégories formant le plus haut niveau de la hiérarchie constitue le premier ensemble de catégories sur lequel la technologie de catégorisation est déployée. L'ensemble de toutes les sous-catégories, descendants directs, de chacune de ces catégories définit un nouvel ensemble courant de catégories, et ainsi de suite jusqu'aux feuilles de la hiérarchie. La catégorisation d'un nouvel exemple se fait en appliquant récursivement les classifieurs obtenus, le résultat d'un classifieur, dans un niveau intermédiaire, servant à sélectionner le classifieur suivant. Cette approche soulève toutefois un certain nombre de problèmes. Tout d'abord, la sélection de l'ensemble courant de catégories est fondée sur des critères pratiques et n'a aucune raison d'être optimale. De plus, dans la mesure où la probabilité de commettre une erreur est *a priori* plus élevée pour des cascades longues (c'est-à-dire comportant de nombreux classifieurs) que pour des cascades courtes, la probabilité d'erreur de cette approche peut être importante si la profondeur de la hiérarchie l'est. Enfin, le choix d'un seul classifieur pour poursuivre la catégorisation d'un exemple ne permet d'explorer qu'une portion congrue de l'espace de recherche et ne permet pas de revenir sur des erreurs faites à un niveau quelconque.

Il est possible de revenir sur ces défauts en considérant d'une part des ensembles courants plus généraux, et d'autre part en sélectionnant plusieurs classifieurs pour la poursuite de la cascade en catégorisation. Soit c une catégorie feuille de l'ensemble courant de catégories[5]. Nous disons qu'un ensemble de catégories $\mathcal{C}$ descendant de c (c'est-à-dire dont toute catégorie descend de c) est *admissible* si toute catégorie finale descendant de c descend aussi d'au moins une catégorie de $\mathcal{C}$. Nous disons de plus que $\mathcal{C}$ est *calculable* si la technologie de catégorisation peut être déployée sur $\mathcal{C}$ en un temps raisonnable[6]. Tout ensemble de catégories admissible et calculable devient ainsi un nouvel ensemble courant possible. Le choix d'un ensemble admissible et calculable à chaque étape de ce processus définit une cascade de classifieurs, qui sera utilisée en catégorisation. Comme nous l'avons souligné, plusieurs classifieurs peuvent être utilisés pour la poursuite d'une cascade en catégorisation. La figure 1 montre deux ensembles admissibles, l'un constitué par toutes les catégories d'un niveau (trait plein), l'autre par des catégories de niveaux différents (pointillé).

Il est clair que toutes les cascades ne sont pas équivalentes en termes d'erreur de catégorisation et de temps de calcul. Le taux d'erreur d'un classifieur augmente en général avec le nombre de classes sur lesquelles il est appris, ce qui semble suggérer qu'il faut privilégier des cascades mettant en jeu des ensembles courants "petits". En contrepartie, de telles cascades reposent sur un plus grand nombre de classifieurs, ce qui augmente la probabilité d'erreur de la cascade. Le problème qui se pose à nous est donc de savoir comment sélectionner la meilleure cascade,

[5] c n'est pas ici une catégorie feuille de la hiérarchie, c'est-à-fire une catégorie finale, mais seulement une catégorie qui n'a aucun descendant dans l'ensemble courant de catégories.

[6] Le nombre de catégories dans $\mathcal{C}$ est ici borné par les performances des classifieurs considérés. Ce nombre est en général de l'ordre de quelques dizaines, voire de quelques centaines pour certains classifieurs.

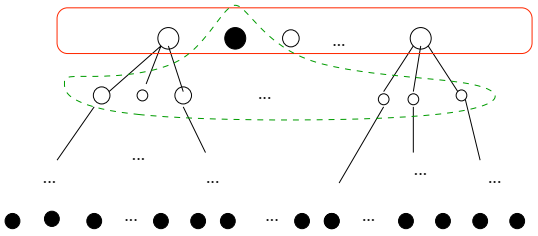

FIG. 1 – Deux ensembles admissibles de catégories. Les catégories finales correspondent aux feuilles de l'arbre.

c'est-à-dire celle qui fournit les meilleures performances en catégorisation tout en permettant de catégoriser un nouvel exemple en un temps raisonnable.

Dans la mesure où le nombre de cascades possibles est potentiellement très élevé (de l'ordre de 2^{p^l} où p désigne le nombre moyen de descendants directs d'une catégorie, et l la profondeur définie par la calculabilité des ensembles de catégories), une recherche exhaustive est ici exclue. Il est donc nécessaire d'explorer de façon efficace le graphe des cascades possibles par le biais d'algorithmes de type *branch-and-bound*, en associant à chaque cascade un coût lié à sa probabilité d'erreur. Pour mettre en œuvre cette stratégie, nous voulons développer les actions suivantes :

1. **Action 1** Etablir des bornes supérieures ou des estimations de la probabilité d'erreur de la meilleure cascade que l'on peut obtenir à partir de n'importe quelle catégorie de la hiérarchie. Il faut noter ici que ces bornes ou estimations dépendent, entre autre, de la technologie de catégorisation considérée.
2. **Action 2** Etablir des algorithmes de type *branch-and-bound* exploitant les possibilités offertes par la parallélisation, et permettant d'explorer de façon efficace l'ensemble des cascades sur la base des bornes et estimations ci-dessus[7].
3. **Action 3** Evaluer les algorithmes ci-dessus sur (a) un jeu de données issu de DMOZ, comprenant environ 100 000 catégories et plusieurs millions de documents[8], et (b), dans la mesure du possible, un jeu de données issu d'un office de brevets.

Nous présentons maintenant le deuxième axe que nous comptons aborder pour résoudre le problème de la catégorisation dans les grands systèmes de catégories.

2.2 Approche directe

L'autre approche que nous voulons étudier dans ce projet vise à échantillonner l'ensemble des catégories finales, à apprendre des classifieurs sur chacun des échantillons, puis à sélectionner la ou les catégories finales sur la base d'un vote entre les classifieurs. Cette approche s'inspire en partie des forêts aléatoires (*random forests*, [8, 3]) qui utilisent un ensemble d'arbres de décision pour déterminer la ou les catégories d'un document. Dans les forêts aléatoires, à chaque nœud

[7]Un cas intéressant ici est celui de la cascade de longueur minimale (c'est-à-dire qui utilise le moins de classifieurs). Cette cascade est optimale sous l'hypothèse que les taux d'erreur d'un classifieur sur les différents ensembles de catégories sont du même ordre.

[8]Ce jeu de données a déjà été constitué par l'un des partenaires du projet.

d'un arbre de décision, un sous-ensemble des attributs est sélectionné aléatoirement, sous-ensemble pour lequel un échantillon bootstrap des exemples sera utilisé, à la fois en apprentissage et en test. Il importe ici d'évaluer plus précisément le comportement des forêts aléatoires lorsque des sous-ensembles d'exemples ou de catégories sont considérés, et de déterminer un algorithme qui soit à la fois précis et rapide dans ces cas. Bien sûr, les différents classifieurs participant au vote doivent être lancés en parallèle pour optimiser les performances du système de catégorisation. Les actions que nous voulons aborder ici sont donc les suivantes :

1. **Action 4** Généraliser les résultats théoriques obtenus sur les forêts aléatoires au cas où catégories et exemples sont sous-échantillonnés.
2. **Action 5** Sur la base de ces résultatst, établir des algorithmes permettant une catégorisation directe dans l'ensemble des catégories finales, déployer ces algorithmes sur une grille de calcul et les évaluer sur les données de test mentionnées ci-dessus.

Références

[1] T. Amund and E. Havard. Parallelization of the incremental proximal support vector machine classifier using a heap-based tree topology. In *Proceedings of the ECML-PKDD Workshop on Parallel and Distributed computing for Machine Learning*, 2003.

[2] S. Berchtold, C. Bohm, and H.-P. Kriegel. The pyramid-technique : Towards breaking the curse of dimensionality. In *Proceedings of SIGMOD'98*, 1998.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1), 2001.

[4] G. C. Cesa-Bianchi N., Conconi A. Regret bounds for hierarchical classification with linear-threshold functions. In *Proceedings of the International Conference on Learning Theory COLT'04*, 2004.

[5] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of svms for very large scale problems. In *Advances in Neural Information Processing Systems*, 2002.

[6] E. Gaussier, C. Goutte, K. Popat, and F. Chen. A hierarchical model for clustering and categorising documents. In *Proceedings of the 24th BCS-IRSG Colloquium on IR Research (ECIR'02)*, LNCS. Springer, 2002.

[7] M. R. Guarracino, C. Cifarelli, O. Seref, and P. M. Pardalos. A parallel classification method for genomic and proteonomic problems. In *Proceedings of 20th International Conference on Advanced Information Networking and Applications (AINA 2006) Volume 2, IEEE Press*, 2006.

[8] T. Ho. Random decision forest. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1995.

[9] J. Lafferty, A. McCallum, and P. F. Conditional random fields : robabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conferernce on Machine Learning, ICML'01*, 2001.

[10] J. Lafferty, X. Zhu, and L. Y. Kernel conditional random fields : representation and clique selection. In *Proceedings of the International Conferernce on Machine Learning, ICML'04*, 2004.

[11] J. Shafer, R. Agrawal, and M. M. Sprint : A scalable parallel classifier for data mining. In *Proceedings of the 22nd VLDB Conference*, 1996.

[12] V. Sindhwani and S. Sathiya Keerthi. Large scale semi-supervised svms. In *Proceedings of SIGIR'06*, 2006.

[13] B. Taskar. Learning structured prediction models : A large margin approach. *PhD Thesis, Stanford University*, 2004.

[14] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine leaning for interdependent and structured output spaces. In *Proceedings of the International Conferernce on Machine Learning, ICML'04*, 2004.

[15] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Machine Learning Research*, 6, 2005.

[16] A. Vinokourov and M. Girolami. A probabilistic hierarchical clustering method for organizing collections of text documents. In *Proceedings of the 15th International Conference on Pattern Recognition*, 2002.

3 Partenaires du projet

Le tableau suivant résume l'information relative aux partenaires de ce projet.

	Etablissement	Grade	Laboratoire	Equipe
Anestis ANTONIADIS	UJF	PR	Lab. J. Kuntzmann	SMS
Michel BURLET	UJF	MCF	Lab. G-SCOP	OC
Yves DENNEULIN	INPG	MCF	Lab. d'Informatique de Grenoble	MESCAL
Eric GAUSSIER	UJF	PR	Lab. d'Informatique de Grenoble	MRIM

Anestis Antoniadis, de l'équipe *Statistiques et Modélisation Stochastique*, supervisera les aspects statistiques du projet. Michel Burlet, de l'équipe *Ompitmisation Combinatoire*, supervisera les aspects liés à l'optimisation combinatoire et la recherche opérationnelle. Yves Denneulin, de l'équipe *Middleware Efficiently Scalable*, supervisera les aspects liés à la parallélisation et le déploiement sur grille de calcul. Eric Gaussier, de l'équipe *Modélisation et Recherche d'Information Multimédia*, supervisera les aspects liés à l'apprentissage automatique et coordonera l'ensemble du projet.