Forêts aléatoires pour l'apprentissage de données massives

R. Genuer[a] and **J.M. Poggi**[b] and **C. Tuleau-Malot**[c] and **N. Villa-Vialaneix**[d]

[a]INRIA, SISTM team & ISPED, INSERM U-897
Univ. Bordeaux, France
robin.genuer@isped.u-bordeaux2.fr

[b]LMO, Univ. Paris-Sud Orsay & Univ. Paris Descartes
Paris, France
jean-michel.poggi@math.u-psud.fr

[c] Lab. Jean-Alexandre Dieudonné, Univ. Nice
Sophia Antipolis, France
malot@unice.fr

[d]MIAT, Université de Toulouse, INRA
31326 Castanet-Tolosan cedex, France
nathalie.villa@toulouse.inra.fr

Mots clefs : Forêts aléatoires, Données massives, Bag of Little Bootstrap, Map Reduce

Introduction

Les forêts aléatoires [1] sont une méthode d'apprentissage largement utilisée dans le cadre de problèmes de régression ou de classification en raison de leur flexibilité et de leurs bonnes performances de prédiction.

Basée sur une approche par « bagging », la méthode est naturellement parallélisable et a donc rapidement été utilisée pour traiter de gros volumes de données. Des adaptations des forêts aléatoires ont ainsi été proposées dans le contexte du « big data ». En particulier, une de ces approches est basée sur le paradigme « Map Reduce » (et est implémentée dans la librairie Mahout [1]) et une autre est l'approche « Bag of Little Bootstrap » [3] qui est en fait définie pour toute méthode basée sur le bootstrap et s'adapte donc de manière immédiate aux forêts aléatoires.

Une étude par simulation avec R

Dans cette proposition de communication, nous présentons les résultats de simulations comparant les diverses approches permettant d'utiliser l'algorithme de forêts aléatoires dans le contexte de données massives. Nous utilisons uniquement R et avons reprogrammé l'ensemble de ces approches en utilisant uniquement le package original **randomForest** [2]. Ce choix a été dicté par la volonté d'obtenir des temps de calcul comparables, indépendants d'une implémentation particulière. Les approches que nous comparons sont (voir [2] pour plus de détails) :

i) l'approche directe (séquentielle) des forêts aléatoires classiques de Breiman,

ii) l'utilisation des forêts aléatoires sur un sous-échantillon de taille réduite (nommée « sampling-RF »),

iii) l'adaptation de l'approche « Bag of Little Bootstrap » au cas des forêts aléatoires (nommée « BLB-RF »),

iv) l'utilisation de Map Reduce pour les forêts aléatoires (nommée « MR-RF »).

Nous utilisons un jeu de données simulé constitué de 15 millions d'observations, généré à partir d'un modèle décrit dans [4]. Il s'agit d'un problème de classification binaire avec une variable à prédire $Y \in \{-1, 1\}$ et 7 variables explicatives X^j, dont 6 sont des « vraies » variables et les autres sont des variables de bruit. Le jeu de données généré, sauvegardé au format « texte simple », a une taille de 1.9Go.

1. https://mahout.apache.org
2. https://cran.r-project.org/web/packages/randomForest

Résultats

Une partie des résultats obtenus sont résumés dans le tableau ci-dessous. Dans chaque ligne, le nom de la méthode est suivi par la valeur des paramètres associés : pour « sampling-RF », le taux d'échantillonnage est donné, pour « BLB-RF », il s'agit du nombre de sous-échantillons K et du nombre d'arbres construits sur chaque sous-échantillon. Enfin, pour « MR-RF » on précise le nombre de d'ensembles dans la partition du jeu de données global ainsi que le nombre d'arbres construits sur chacun de ces ensembles.

Méthode	Temps d'exec.	BDerrForest	errForest	errTest
Breiman's RF	7 h	4.6	4.6	4.4
sampling-RF 10%	3 min	4.6	4.4	4.3
sampling-RF 1%	9 sec	4.6	4.4	4.4
sampling-RF 0.1%	1 sec	5.6	4.7	4.6
sampling-RF 0.01%	0.3 sec	4.7	6	5.7
BLB-RF 5/20	1 min	4.1	4.3	4.3
BLB-RF 10/10	3 min	4.1	4.3	4.3
MR-RF 100/1	2 min	14	4.2	4
MR-RF 100/10	2 min	8.6	4.1	4.3
MR-RF 10/10	6 min	8.5	4.3	4.2
MR-RF 10/100	21 min	4.5	4.2	4.3

TABLE 1 – Performances des diverses approches de forêts aléatoires. On trouve, en colonne 1, le nom de la méthode suivi du paramètre d'intérêt, le temps d'exécution en colonne 2, le taux de mauvais classement multiplié par 10^3), l'erreur OOB version données massives (col. 3), l'erreur OOB utilisant tout le jeu de données (col. 4) et une erreur calculée sur un échantillon test (col. 5).

Conclusion

Les approches considérées donnent des résultats satisfaisants tant pour le temps de calcul (gain très significatif par rapport à la version séquentielle) que pour le taux d'erreur test. Cependant, il faut noter que les performances de « sampling-RF » se dégradent lorsque le taux d'échantillonnage devient vraiment trop faible. De plus, le principal problème rencontré avec « MR-RF » est que l'estimation de l'erreur OOB calculée uniquement sur les sous-ensemble (qui est celle naturellement fournie par la méthode) est très mauvaise. En outre (résultats non founis), nous avons mis en valeur des biais d'estimation possibles dans le cas de la version « MR-RF » que nous présenterons et commenterons : ceux-ci nous montrent que les approches « Big Data » des forêts aléatoires doivent être utilisées avec attention et que des améliorations de celles-ci, utilisant par exemple des pondérations adéquates ou bien des versions « en ligne », sont envisageables.

Références

[1] L. Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[2] R. Genuer, J. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix. Random forests for big data. Preprint arXiv :1511.08327. Submitted for publication.

[3] A. Kleiner, A. Talwalkar, P. Sarkar, and M.I. Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 76(4) :795–816, 2014.

[4] J. Weston, A. Elisseff, B. Schoelkopf, and M. Tipping. Use of the zero norm with linear mmodel and kernel methods. *Journal of Machine Learning Research*, 3 :1439–1461, 2003.