Discussion of Parallel Construction of Decision Trees with Consistently Non- Increasing Expected Number of Tests

Servane Gey, Jean-Michel Poggi

Discussion of Parallel Construction of Decision Trees with Consistently Non-Increasing Expected Number of Tests

Servane Gey [1,3], and Jean-Michel Poggi [2]

[1] MAP5, University Paris Descartes France
[2] LMO, University Paris Sud, Orsay, France
[3] Corresponding author, Jean-Michel Poggi, e-mail Jean-Michel.Poggi@math.u-psud.fr

1. Introduction

We would like to congratulate the authors for their interesting paper in which they propose a new parallel construction of decision trees and show how their new method relates to other methods that have been previously proposed. We found the paper to be stimulating and found that it offers insightful ideas related to the new construction.
The algorithm permits to remove false alarms occurring at the beginning of the tree, thus reduces the number of tests. Indeed, the construction of a decision tree needs in general to compensate early false alarms by adding more tests leading to deeper trees. It is part of the renewed interest for tree-based methods; we can indeed quote numerous survey papers, as for example Patil and Bichkar (2012) in computer science or Loh (2014) in statistics, which contain useful extensive bibliographies.
The paper is clearly motivated by applications and implementation issues. As statisticians, we will try to highlight this contribution from a statistical perspective, so our comments center on questions related to how to generate and how to select tree models.

2. About using decision trees for time series segmentation

It may be interesting to use decision trees for time series segmentation, in particular when the number of observations is huge. Indeed, an exhaustive search is computationally too greedy in this case, whereas algorithms like CART or C4.5 are well adapted. The main issue of the use of decision trees is false alarms, which cannot be removed if appearing at the beginning of the tree. Gey et al. (2008) proposed an hybrid algorithm which uses CART to preselect a family of change points, and then removes false alarms with an exhaustive search. The main drawback of this hybrid algorithm is that the final predictor's configuration is no more a tree. Hence SF-GOTA is a good alternative in this case since it removes false alarms and provides a tree at the same time.

3. About model selection and decision trees

3.1 L-curve

As shown in Figures 12 to 15 of the paper, ENT is a decreasing function of the parameter s in SF-GOTA having an « L » form. This is a classical good behaviour in the model selection field when one aims at selecting the optimal value of some calibration parameter occuring in the complexity penalty.

Hence, a good alternative to select the optimal value of s could be to use L-curve heuristic (as proposed in Hansen (1992), Engl et al. (1994) or Hanke (1996)) : find the point of the L-curve for which there is a corner, and take the corresponding value of s.

3.2 Connexion between misclassification rate and ENT minimization

The authors draw on ENT to select optimal trees in SF-GOTA. Nevertheless, even if ENT is directly relied to the tree's complexity, taking only ENT into account means forgetting the important compromise to be made between misclassification rate and complexity. It is shown in Gey et al. (2014) that this compromise is necessary to obtain good decision tree predictors. The authors claim that the tree selected at the end of SF-GOTA makes this compromise, but with s ranging from 1 to at most 6, the trees provided by SF-GOTA are undeep by construction. Then it could be interesting to compare prediction performance between trees provided by SF-GOTA and trees provided by C4.5 or CART to quantify the loss of prediction performance with respect to the huge gain of computational time.

4. About generating tests using trees or forests

The next remark is related to what can be understood as a restriction for the application since the discussed paper focuses on binary datasets to illustrate the value of the proposed algorithm. Indeed, starting from data described by an array of the classical form individuals x variables, one can use a CART model, which includes an optimal pruning, to automatically generate splits, that is a variable and a threshold value (for numerical variables) or a subset of values (for categorical variables). If the dataset is too large, this can applied to a smaller pilot sample. Then the selected splits can be used as candidates to generate tests and to encode the data as done in the paper. More generally, Random Forests and related methods can help to rank the variables (see Genuer et al. (2010)) and construct the former tests according to the most important ones or by selecting splits generated according to the previous strategy.
Alternative ideas could be to consider deterministic splits on each variable as performed in dyadic trees (see Blanchard et al. 2007) or purely RF where random splits are selected (see Biau et al (2008)), to construct the list of tests.

5. A final remark about Big Data issues

Many tree-based schemes including bagging and Random Forests admit reformulations based on Hadoop programming (see Prajapati (2013)). One of the issues discussed at the beginning of the discussed paper relates to Big Data, which is one of the motivations of such proposals. The paper mainly addresses the massive data issue but the problem of how to manage online updating of trees, which is the second characteristic of Big Data problems, is not considered. It could be of interest to explore this potential source to speed up the data processing, see for example Gama et al. (2003) for adapting decision trees to high-speed data streams.

References

Biau, G., Devroye, L., Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *The Journal of Machine Learning Research*, *9*, 2015-2033.

Blanchard, G., Schäfer, C., Rozenholc, Y., Müller, K. R. (2007). Optimal dyadic decision trees. *Machine Learning*, *66* (2-3), 209-241.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Engl, HW, Grever, W (1994). Using the L-curve for determining optimal regularization parameters. In *Numerische Mathematik, 69*, pp. 25-31. Springer.

Gama, J., Rocha, R., & Medas, P. (2003). Accurate decision trees for mining high-speed data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '03). ACM, New York, USA, 523-528.

Gey, S., Mary-Huard T., (2014). "Risk Bounds for Embedded Variable Selection in Classification Trees." *Information Theory, IEEE Transactions on* vol. 60 (3), pp. 1-12.

Gey, S., Lebarbier, E. (2008). Using CART to Detect Multiple Change-Points in the Mean for Large Samples. Research Report No.12, Statistics for Systems Biology (SSB) Preprint.

Genuer, R., Poggi, J. M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*(14), 2225-2236.

Hanke, M. (1996). Limitations of the L-curve method in ill-posed problems. *BIT Numerical Mathematics*, *36*(2), 287-301.

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the L-curve. *SIAM review*, *34*(4), 561-580.

Loh, W.-Y. (2014), Fifty Years of Classification and Regression Trees. *International Statistical Review*. doi: 10.1111/insr.12016.

Patil, D. V., Bichkar, R. S. (2012). Issues in optimization of decision tree learning: A survey. *International Journal of Applied Information Systems (IJAIS)*, 3(5), 13-30.

Prajapati, V. (2013). *Big Data Analytics with R and Hadoop*. Packt Publishing Ltd.