

R-implementation of the TREERANK algorithm

Nicolas Baskiotis

NICOLAS.BASKIOTIS@CMLA.ENS-CACHAN.FR

CMLA UMR CNRS No. 8656

ENS Cachan

61, avenue du président Wilson, Cachan, 94235, France

Stéphane Cléménçon

STEPHAN.CLEMENCON@TELECOM-PARISTECH.FR

Institut Telecom LTCI UMR Telecom ParisTech/CNRS No. 5141

Telecom ParisTech

46 rue Barrault, Paris, 75634, France

Marine Depecker

MARINE.DEPECKER@TELECOM-PARISTECH.FR

Institut Telecom LTCI UMR Telecom ParisTech/CNRS No. 5141

Renault DREAM-DTAA

Guyancourt, 78288, France

Nicolas Vayatis

NICOLAS.VAYATIS@CMLA.ENS-CACHAN.FR

CMLA UMR CNRS No. 8656

ENS Cachan

61, avenue du président Wilson, Cachan, 94235, France

Editor:

Abstract

TREERANK is a R-package for building tree-based ranking rules through the tree induction method proposed in Cléménçon and Vayatis (2009). It also comprises a list of routines for reading or visualizing ranking trees. We provide an easy-to-use GUI in order to perform model selection and interpret the results produced by this learning algorithm, specifically designed for ROC optimization.

Keywords: bipartite ranking, ROC optimization, TREERANK algorithm, tree induction method, R-package

1. Introduction

Bipartite ranking, also termed *nonparametric scoring*, is an ubiquitous issue, one may find in anomaly detection, medical diagnosis, credit-risk screening, or information retrieval for instance. Indeed, in a wide variety of fields, practitioners need to learn scoring/ranking functions for discriminating between two populations from multivariate data with binary labels. Their performance is generally evaluated through ROC analysis in a visual manner, see Fawcett (2006) and the references therein. ROC curves and related summary statistics such as the celebrated AUC criterion are indeed among the most popular tools for measuring ranking accuracy. Whereas research in the field of bipartite ranking has essentially focused on theoretical aspects such as the extension of the Empirical Risk Minimization principle, the majority of statistical techniques for learning ranking rules rest either on the modelling of the posterior probability or else on combining binary classifiers in an ad-

ditive fashion. However, a novel tree induction procedure, entirely tailored for bipartite ranking and producing, from a training sample, a piecewise constant scoring function of which ROC curve mimics the behavior of an adaptive linear-by-part interpolant of the optimal ROC curve has recently been proposed and thoroughly studied in a series of papers, see Cl  men  on and Vayatis (2009, 2008). The software we present here implements this statistical learning algorithm, named TREERANK. It thus enables ROC optimization and outputs a list of indicators describing the ranking model thus learnt and its statistical performance. It must be run under R, a free software environment for statistical computing and graphics, see <http://www.r-project.org/>. The related package is available at <http://treerank.sourceforge.net/>.

2. Tools and Functionalities

We first describe the main functionalities of the TREERANK R-package.

2.1 Learning to Rank

Bipartite ranking involves sorting all possible values x for an input variable X , valued in a feature space $\mathcal{X} \subset \mathbb{R}^d$ and to which is assigned a random binary label $Y \in \{-1, +1\}$, by order of magnitude of the posterior probability $\mathbb{P}\{Y = +1 \mid X = x\}$. In practice, rankings are defined through *scoring functions* $s : \mathcal{X} \rightarrow \mathbb{R}$ and their accuracy is measured in terms of ROC curve. Based on a collection $\{(x_i, y_i) : 1 \leq i \leq n\}$ of independent realizations of the random pair (X, Y) , the learning algorithm in question produces models that can be easily summarized in the form of an oriented rooted binary tree graph, see Fig. 1. The ranking can be directly read by perusing the terminal leaves from the left to the right. As for other tree-based learning methods, a greedy top-down recursive partitioning strategy is first implemented, leading to a *Master Ranking Tree*. Each split corresponds itself to a tree-based rule, obtained through a cost-sensitive version of the standard CART methodology (see Breiman et al. (1984)) we call LEAFRANK, the cost being taken equal to the rate of positive instances within the node to split in order to maximize the AUC criterion recursively. A pruning procedure possibly follows the growing stage, where children of a same parent node are recursively merged so as to maximize a cross-validation based estimate of the AUC criterion. A crucial advantage of such a tree-structured recursive partitioning method lies in its ability to handle qualitative predictor variables (up to a dummy coding) and incomplete data in both the training samples and future observations to be predicted.

2.2 Model Interpretation

A list of statistics describing the ranking model is readily available, using clicking-through functionalities of the GUI: for each node, one may visualize the number of positive instances (respectively, negative instance) lying in it, the classification tree corresponding to its split and the gain it induces (in terms of AUC). Interpretability is one of the key advantages of tree-based ranking rules with axis parallel splits. In this respect, together with the ranking tree schematic, a graphic showing the relative importance of each predictor variable in the ranking rule is automatically displayed. In regards to ranking performance, an estimate of the ROC curve of the scoring function output by the algorithm based on a test sample

(when available) is plotted, completing the overoptimistic estimate provided by the training curve.

2.3 Tuning Ranking Trees

In order to run the TREERANK algorithm, the user has to pick several tuning parameters with the help of the GUI: the maximum depth of the master ranking tree, the minimum number of data lying in a cell, the possible use of pruning (and, in this case, the number of replications in the cross validation procedure) essentially, as well as similar characteristics for the LEAFRANK splitting procedure. She/he must also choose which of the two labels is considered as the positive one. In addition, submodels of the original tree-based ranking rule can be selected at hand (and saved) by clicking on the internal nodes that one would like to see as terminal, the ROC curve estimate being synchronously modified.

2.4 Prediction

Given a (stored) ranking tree, loading new input data (with no labels), a dedicated function implements the corresponding ranking rule in order to sort/score the data. The whole prediction process can be carried out using the GUI.

2.5 The Two-Sample Problem

The ranking method can be used for solving two-sample problems, *i.e.* testing in a non-parametric fashion whether two samples are drawn from the same distribution or not, see Cl  men  on et al. (2009). The procedure is performed in two stages, by splitting the data into two subsets, the choice of their sizes being left to the user. The first sample is used for implementing the ROC optimization procedure, while a standard Mann-Whitney-Wilcoxon test (MWW), at a level fixed by the user, is applied to the second dataset, once sorted using the scoring function learnt precedingly.

2.6 Modularity

TREERANK has been conceived in a very modular way, so that cost-sensitive versions of any classification method may serve as splitting rule in the meta-algorithm (in the present version, cells can be split using SVM too). With the aim to reduce instability and/or increase ranking accuracy, feature randomization and bootstrap aggregation techniques can also be incorporated, see Cl  men  on (2010).

3. Availability and Requirements

Installation of the TREERANK package requires version at least 2.9.2 of the R-statistical software, as well as preliminary installation of the following R-packages, all available at <http://CRAN.R-project.org>: TCLTK, IGRAPH and COLORSPACE for graphical purposes, KERNLAB and RPART when considering SVM-based splits and CART-based splits respectively and COIN for the MWW test. Refer to Therneau et al. (2009), Hothorn et al. (2008) and Karatzoglou et al. (2004). The R-package TREERANK is released under the GPL license. All source codes follow the **S3** standard in R. Files in TREERANK include source

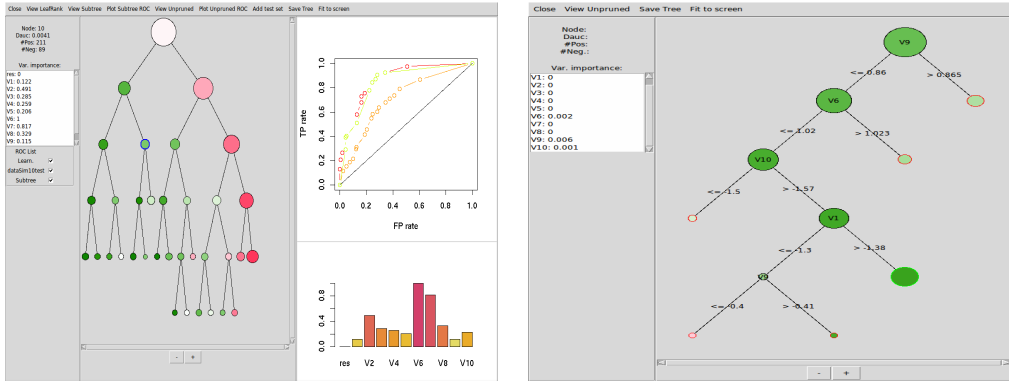


Figure 1: Fitting a tree-structured ranking rule via the TREERANK R-package.

files, a quick-start guide with compilation instructions, a tutorial to get started with the software, as well as a collection of data examples.

Acknowledgments

We would like to acknowledge support for this project from the ENS Cachan.

References

- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.
- S. Cléménçon. Ranking Forests. Technical Report hal-00452577, HAL, 2010. URL <http://hal.archives-ouvertes.fr/hal-00452577/fr/>.
- S. Cléménçon and N. Vayatis. Tree-structured ranking rules and approximation of the optimal ROC curve. In *Proceedings of ALT 2008*. Springer, 2008.
- S. Cléménçon and N. Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, 2009.
- S. Cléménçon, M. Depecker, and N. Vayatis. AUC optimization and the two-sample problem. In *Advances in Neural Information Processing Systems*, 2009.
- T. Fawcett. An Introduction to ROC Analysis. *Pattern Recog. Lett.*, 27(8):861–874, 2006.
- T. Hothorn, K. Hornik, M.A. van de Wiel, and A. Zeileis. Implementing a class of permutation tests: The coin package. *Journal of Statistical Software*, 28(8):1–23, 2008.
- A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. kernlab an s4 package for kernel methods in r. *Journal of Statistical Software*, 11(9):1–20, 2004.
- T.M. Therneau, B. Atkinson, and B.D. Ripley. rpart: Recursive partitioning. Technical report, 2009. URL <http://CRAN.R-project.org/package=rpart>.