Pruning Gradient Boosted Oblivious Decision Trees

A. Rogozhnikov, A. Panin

Abstract

This paper describes an approach to the problem of pruning the Gradient Boosted Oblivious Decision Trees that aims in maintaining the maximal quality of the full ensemble while removing most of the trees from it and adjusting the individual trees. The results are then evaluated by the ROC AUC score on the HIGGS dataset.

I. Introduction

One of the most popular method branches used in state of art machine learning applications is the inductive decision tree learning [1]. While the greedy decision tree composition methods are relatively computationaly inexpensive, they tend to show remarkable performance on datasets with numerous features of different origin. The decision trees are also used to perform feature selection [2] from the high dimensionality data for further use in more sophisticated models. One more obvious advantage of decision trees is their interpretability, as compared to most other approaches.

Apart from the classical decision trees, there are several known alternative decision tree forms, and among them, the so called Oblivious Decision Trees (ODT) [3], that differ from classical ones in that they use the same splitting criterion in every node within a depth layer, thus in effect being decision tables.

This simplification, while diminishing the individual efficiency of a trained ODT model, allows for a considerable speed up in the training algorithm runtime, thus increasing performance in ensemble models (typically bagging [4][5] or boosting[6][7] variations), that average the results from sets of decision trees to get the best quality.

II. On Gradient Boosting

The majority of approaches, suggested in the current paper, were inspired by and tested on the so called Gradient Boosted Decision Trees [7] ensemble model, that can be formally defined as a result of additive iterative gradient optimisation of the loss function.

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

$$F_m(x) = F_{m-1}(x) + \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + f(x_i))$$

Fig.1 the formal gradient boosting definition

Here, F stands for the ensemble model, L is the loss function, H represents the model space (in this case, decision trees), f stands for a particular model within H, yi is the label of the ith training sample, n âĂŞ total amount of samples, m âĂŞ amount of gradient boosting optimization steps.

It should be of course noted, that the practical implementation of gradient boosting is swarming with countless heuristics from the domain of divine providence and/or the black magic. The above formulae are only presented to demonstrate the core principle of gradient boosting.

The advantages and drawbacks of the gradient boosted ODT are numerous, yet their analysis lies so far outside [8] the discourse universe of this article. What is more important to it, some practical applications demand that the decision making time of the final model stays low enough. Unless some other structurally faster approach is possible, the ensemble needs to be pruned efficiently to reach the required execution speed.

III. Application example

One such practical application might be the problem of binary signal/background classification of the event (particle collision) within the LHCb experiment [9]. LHC collides a bunch of protons every 50 nanoseconds, thus producing a gigantic influx of data (approx. $2 * 10^7$ events per second). The main objects of interest within the experiment are the very rare decays, while the rest of the data is generally uninteresting.

In order to filter out some of the obviously background events, LHC uses the multi-level trigger architecture, that filters out most of the data flow on each level. The low level triggers are removing the obvious background events, while the higher levels are increasing the precision even further by dealing with the events, approved by the lower level triggers, thus dealing with harder problem on smaller (yet still immense) data flow.

The possible application of Gradient Boosted ODT can be seen within the architecture of high level trigger within the LHCb detector.

References

[1] iang Yang, Philip S. Yu, Zhou Zhihua, and David Hand et al "Top 10 algorithms in data mining", Knowledge and Information Systems 14.1: 1-37, 2008

[2] . Deng, G. Runger, "Feature Selection via Regularized Trees", Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, 2012

[3] ohavi, R. & Li, C. (1995). Oblivious decision trees, graphs, and top-down pruning. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (pp. 1071âĂŞ1077). San Mateo, CA: Morgan Kaufmann.

[4] reiman, L. (1996a). Bagging predictors. Machine Learning 26 (2), 123âĂŞ140.

[5] reiman, L. Random forests. Machine Learning, 45(1):5âĂŞ32, 2001

[6] reund, Y. & Schapire, R. (1996). Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, 148âĂŞ156

[7] riedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." (February 1999)

[8] . Augusto Alves Jr. et al. (LHCb Collaboration) (2008). "The LHCb Detector at the LHC". Journal of Instrumentation 3 (8): S08005. Bibcode:2008JInst...3S8005T. doi:10.1088/1748-0221/3/08/S08005. (Full design documentation)