Additive Groves with very simple features for brain fibers classification

Daria Sorokina
School of Computer Science
Carnegie Mellon University
daria@cs.cmu.edu

Alexander Sorokin
Department of Computer Science
University of Illinois at Urbana-Champaign
sorokin2@uiuc.edu

Abstract

In this document we describe the entry that came in the third place in the "Supervised Learning - Match Expert" challenge of the ICDM'09 Data Mining Contest. The main method used in our approach is a recently developed tree ensemble algorithm called Additive Groves.

Code for Additive Groves is publicly available on the author's website.

1. Supervised Learning Challenge

The data for this competition described scans of three different human brains. Each data point referred to a single fiber in the brain. An expert has labeled eight fiber tracks in one of the brain; the goal was to predict correct labels for the same fiber tracks in the two other brains.

2. Feature Extraction

Each fiber was initially represented as a sequence of points in 3D space. The brains were normalized by the organizers of the challenge, so we did not perform any alignment or scaling of the data.

We extract simple features to describe location of fibers in the brain. In particular we considered fiber length, starting location, ending location, bounding box of the fiber points and finally the center of mass of all points of the fiber. To compute the length of the fiber, we connected each two adjacent points with a line and summed the lengths of the line segments (1 feature). The starting and ending location are the coordinates of the first and the last points in the fiber (6 features). The bounding box of the fiber points is computed as minimum and maximum for each dimension (x,y,z) among all points in the fiber (6 features). The center of mass is taken as the mean value of fiber point coordinates (3 features).

These features describe very rough location of fibers in the brain. This representation is very compact, as we use only 16 real numbers to describe each fiber. Code and extracted features are freely available upon request.

3. Additive Groves

Additive Groves [2] is a tree ensemble algorithm based on regression trees, additive models and bagging and is capable of both fitting additive structure of the problem and modelling nonlinear components with large trees at the same time. Combination of these properties makes it superior in performance to other existing tree ensemble methods like bagging, boosting and Random Forests.

Additive Groves consists of bagged additive models, where every element of an additive model is a tree. A single grove is trained similar to an additive model: each tree is trained on the residuals of the sum of the predictions of the other trees. Trees are discarded and retrained in turn until the overall predictions converge to a stable function. In addition, each grove of a non-trivial size is iteratively built from smaller models: either a tree is added to a grove with fewer trees or the size of trees is increased in a grove with smaller trees; which of the two steps happen in each particular case is defined by comparing models on out-of-bag data. A single grove consisting of large trees can and will overfit heavily to the training set, therefore bagging is applied on top in order to reduce variance.

We refer the reader to the paper where Additive Groves were introduced [2] for more detailed description of the algorithm.

In this challenge, we were using the classification variant of the algorithm [1] where the optimal param-

eters are chosen based on ROC score on the validation set.

4. Training models

The first model we built was separating the eight labeled fiber tracts from the rest of the data. The amount of training data describing "label 0" fibers was unnecessarily large, so we have applied the following approach to pull out interesting training data set of reasonable size:

1. Pull out a random initial training set of acceptable size. We have opted for 10000 data points from eight labeled fiber tracts and 10000 "label 0" data points.
2. Build a model
3. Generate predictions for all data points of labeld brain that did not get into the training data
4. Take all data points where the model made mistakes or was unsure (predicted a value between 0.3 and 0.7), add them to the training data.
5. Repeat from step 2 until all predictions are correct.

We had to perform only three iterations of this process and ended up using only about 10% of the whole data for training the final model. After that we trained eight smaller models: each was predicting whether a fiber belongs to one of the eight tracts. We used only fibers from the eight labeled tracts for training.

5. Predictions

First, we predicted whether the fibers belong to any of the eight tracts using the first model that we trained. We have chosen the top 25000 fibers (approximately equals the number of fibers in the eight tracts of the first brain) for each brain to consider for further classification and discarded the rest.

After that we have predicted for each fiber the probability of being in each of eight tracts and labeled it with the class that had the highest probability. We further tried to make use of the scoring technique in this challenge (up to three clusters could match each class) and created several submissions by clustering fibers in each predicted class in different ways.

6 TreeExtra package

The main classification algorithm used in this entry — Additive Groves — is available as a part of TreeExtra package. TreeExtra is a set of command line tools written in C++/STL. You can find binaries, code and manuals at

www.cs.cmu.edu/~daria/TreeExtra.htm

References

[1] D. Sorokina. Application of Additive Groves Ensemble with Multiple Counts Feature Evaluation to KDD Cup'09 Small Data Set. In *Proceedings of the KDD Cup workshop*, 2009.

[2] D. Sorokina, R. Caruana, and M. Riedewald. Additive Groves of Regression Trees. In *Proceedings of European Conference in Machine Learning*, 2007.