Tuning

Prof. Eric A. Suess

April 13, 2020

Introduction

In Chapter 11 the author discusses improving model performance.

The idea of **model performance** is discusses in terms of

- Tuning parameters
- Ensembles

Tuning parameters can be used to improve the performance of a single model.

Ensembles can be used to build of team of learners that may have better performance than a single model.

Tuning Parameters

We know a little about **tuning parameters** from the C5.0 algorithm where we introducted **Boosting**.

And from kNN where tried different values of k.

Suppose we set a range of values for a tuning parameter and then fit the model for each of the values of the tuning parameter, keeping a measure of performance. Then we can pick the **best value** of the tuning parameter and the model produced.

caret package

The **caret** package in R gives functions that make tuning a model easy.

The functions train(), trainControl() and expand.grid()

The **kappa** value can be used to **optimize**.

Running experiments

In Machine Learning tuning over a grid is called running an **experiment**.

Ensembles

The author discusses **meta-learners**.

The technique of combinding and managing the predictions of multiple models falls within a wider set of **meta-learning** methods that broadly encompass any technique that involves learning how to learn.

These may include:

- gradually improved performance by automatically iterating over design decisions
- self-modifying and adapting to learning tests

Ensembles

All **ensemble methods** are based on the idea that by combining multiple weaker learners, a stronger learner is created.

Use a **team of models** or a **committee of models**.

- Bagging
- Boosting
- Random Forests

Bagging

Boostrap aggregating or Bagging

A number of training datasets are generated by boostrap sampling the original training data. Boostrap sampling is sampling the same number of rows as there are in the training data, **with replacement**.

These datasets are used to generate a set of models using a single learning algorithm.

The models' predictions are combined using **voting** (for classification) or **averaging** (for prediction).

Bagging neads **unstable** learners. So bagging is often used with decision trees.

Boosting

Boosting uses ensembles of models trained on resampled data (**re-weighted datasets**) and a vote to determine the final classification or average for a prediction.

The resampled datasets in boosting are constructed specifically to generate **complementary learners**, and the vote is weighted based on each model's performance rather than giving each an equal vote.

AdaBoost

AdaBoost or adaptive boosting.

The algorithm is based on the idea that generating weak learners that iteratively learn a larger portion of the **difficult-to-classify** examples in the training data by paying more attention (that is giving more weight) to the often misclassified examples.

Other Boosting Algorithms

- **Gradient Boosting Machines**
- **XGBoost**

See A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning

See

1. XGBoost website.
2. Getting Started
3. Gradient Boosting, Decision Trees and XGBoost with CUDA

Random Forests

This methods combines the base principles of Bagging with **random feature selection** to add additional diversity to the decision tree models.

After the ensemble of trees is generated, the model uses a vote to combine the trees' predictions.

Because random forests use only a small, random portion of the full feature set, it can handle **extremely large datasets**.

Machine Learning Competitions

The author gives an example of picking the most accurate model for submission to a **machine learning competition**.

See the end of Chapter 11.