8

The Boosting Approach to Machine Learning: An Overview

Robert E. Schapire[1]

Summary

Boosting is a general method for improving the accuracy of any given learning algorithm. Focusing primarily on the AdaBoost algorithm, this chapter overviews some of the recent work on boosting including analyses of AdaBoost's training error and generalization error; boosting's connection to game theory and linear programming; the relationship between boosting and logistic regression; extensions of AdaBoost for multiclass classification problems; methods of incorporating human knowledge into boosting; and experimental and applied work using boosting.

8.1 Introduction

Machine learning studies automatic techniques for learning to make accurate predictions based on past observations. For example, suppose that we would like to build an email filter that can distinguish spam (junk) email from non-spam. The machine-learning approach to this problem would be the following: Start by gathering as many examples as possible of both spam and non-spam emails. Next, feed these examples, together with labels indicating if they are spam or not, to your favorite machine-learning algorithm which will automatically produce a classification or prediction rule. Given a new, unlabeled email, such a rule attempts to predict if it is spam or not. The goal, of course, is to generate a rule that makes the most accurate predictions possible on new test examples.

[1] Robert E. Schapire is with AT&T Labs – Research, Shannon Laboratory, 180 Park Avenue, Florham Park, NJ 07932, USA (URL: `www.research.att.com/~schapire`).

D. D. Denison et al. (eds.), *Nonlinear Estimation and Classification*

Building a highly accurate prediction rule is certainly a difficult task. On the other hand, it is not hard at all to come up with very rough rules of thumb that are only moderately accurate. An example of such a rule is something like the following: "If the phrase 'buy now' occurs in the email, then predict it is spam." Such a rule will not even come close to covering all spam messages; for instance, it really says nothing about what to predict if 'buy now' does not occur in the message. On the other hand, this rule will make predictions that are significantly better than random guessing.

Boosting, the machine-learning method that is the subject of this chapter, is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. To apply the boosting approach, we start with a method or algorithm for finding the rough rules of thumb. The boosting algorithm calls this "weak" or "base" learning algorithm repeatedly, each time feeding it a different subset of the training examples (or, to be more precise, a different distribution or weighting over the training examples[2]). Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that, hopefully, will be much more accurate than any one of the weak rules.

To make this approach work, there are two fundamental questions that must be answered: first, how should each distribution be chosen on each round, and second, how should the weak rules be combined into a single rule? Regarding the choice of distribution, the technique that we advocate is to place the most weight on the examples most often misclassified by the preceding weak rules; this has the effect of forcing the base learner to focus its attention on the "hardest" examples. As for combining the weak rules, simply taking a (weighted) majority vote of their predictions is natural and effective.

There is also the question of what to use for the base learning algorithm, but this question we purposely leave unanswered so that we will end up with a general boosting procedure that can be combined with any base learning algorithm.

Boosting refers to a general and provably effective method of producing a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb in a manner similar to that suggested above. This chapter presents an overview of some of the recent work on boosting, focusing especially on the AdaBoost algorithm which has undergone intense theoretical study and empirical testing.

[2] A distribution over training examples can be used to generate a subset of the training examples simply by sampling repeatedly from the distribution.