

CS5339 Project – Active Learning in Classification

Ma Yuan E0674520

March 22, 2021

1 Introduction

The fundamental problem of binary classification is central to both classical and modern machine learning. Many of the earliest works focused on simple classification rules, such as linear classifiers or nearest-neighbor rules, whereas more recent developments have led to more sophisticated ideas such as kernel methods, artificial neural networks, and more. Many such methods can be viewed as *combining simple classifiers* to produce a *complex classifier*.

In this report, we will overview a famous algorithm falling under this category, known as AdaBoost, which was introduced by Freund and Schapire in 1997 [?], and subsequently significantly impacted both the theory and practice of machine learning. The authors were awarded the 2003 Gödel prize, and the algorithm is often considered to be one of the most effective “off-the-shelf” algorithms requiring minimal tuning and tweaking [?, ?].

Before proceeding with a formal description, we outline some of the main high-level concepts and ideas behind AdaBoost:

- The algorithm is given a class of *weak learners*, also known as *base learners*, containing classifiers that are not very powerful in themselves. Despite this, suitably “combining” many such classifiers can lead to a much more powerful classifier.
- The combining is done by a simple voting strategy – each base learner classifies a given point as positive or negative, and the final classification is performed according to a weighted vote of these base learners, where higher votes are given to the base learners that are believed to be more accurate. The role of AdaBoost is to *select the base learners*, as well as choosing the voting weight for each of them.
- The base learners are selected one at a time, and are chosen to minimize a *weighted training error* according to some weights that are maintained on the data points (not to

be confused with the voting weights). The more a point has been classified correctly by previously-selected base learners, the lower weight it receives, whereas a point that was previously classified incorrectly many times is assigned a higher weight. This ensures that more attention is focused on the points where we need to make up for earlier mistakes.

- The weights are updated using the idea of *multiplicative weight updates*, which is a far-reaching tool that has also been explored in numerous other fields [?].

To appreciate the third of these points, it is useful to think of the following analogy: If a teacher wants to teach their students a number of concepts, then they should focus more attention on the concepts that many students scored poorly on in previous tests/exams.

2 Description of AdaBoost

In this section, we formally introduce the AdaBoost algorithm, and give some numerical examples.

2.1 Preliminaries

2.2 The Algorithm

2.3 Experimental Examples

3 Mathematical Analysis

4 Extensions and Further Results

In this section, we briefly outline some more advanced algorithms and theory building on the previous sections. Due to space limitations, we only provide a short discussion on each of these.

4.1 Base Learners Beyond Decision Stumps

4.2 Multi-Class Boosting

4.3 Characterization of the Test Error

Appendix

A Proof of Theorem ?? (AdaBoost Training Error Guarantee)