CS 374 Assignment #8
Computational Learning Theory

Due the week of April 12, 2021

—

This week you will revisit topics that you considered briefly during the first week of the semester: the VC dimension and the PAC (Probably Approximately Correct) framework for analyzing learning algorithms. You will consider issues such as determining the number of examples needed to learn a concept. You will also consider a variety of concept classes and see how they can be proven to be PAC-learnable.

A number of important machine learning algorithms have their roots in the PAC model and, more generally, in the computational learning theory community. Among these is AdaBoost, one implementation of the general framework of Boosting.

What to turn in. For this assignment, you will need to complete four written exercises and implement AdaBoost. Solutions to the four exercises should be written up individually. Please submit the problem set as a pdf. You may implement AdaBoost either alone or with your tutorial partner. Please submit all code in a single compressed file.

What to expect during the meeting. Your job during the meeting will be to formally present the written exercises. Take us through them as if you are teaching a class and using the examples to illustrate the material. Please be prepared to present any/all exercises, as I will determine in the meeting who will present which. Please also be prepared to demonstrate and review your AdaBoost implementation.

1 Learning Theory

1.1 Reading

Please read Mitchell, Chapter 7 (Computational Learning Theory), up to page 220.

1.2 Exercises

Please do the following exercises, which I will ask you to present and turn in.

- From Mitchell, 7.1, 7.3, 7.5 b, 7.7 a & b.

2 AdaBoost

As described above, the AdaBoost algorithm is one of several that have grown out of the PAC learning model. This week you will learn about this algorithm through reading and an implementation exercise. (Next week you will consider Boosting more generally, as well as other methods that involve ensembles of classifiers.)

2.1 Reading

Please read "A Short Introduction to Boosting" by Yoav Freund and Robert E. Schapire.

I expect that you will find this paper to be quite clear, though there are some items on page 6 that you might find a little bit confusing. The basic idea described there is as follows. Since AdaBoost concentrates on training examples that are close to the margin, it naturally increases the margin over time.

Also, note that "OCR" stands for "Optical Character Recognition". The authors assume that this is obvious to the reader, which is reasonable within the machine learning community, but some of you might not yet be familiar with this.

2.2 Implementation

Please implement the AdaBoost algorithm as given on page 3 of the Freund and Schapire paper. The algorithm requires that you train a weak learner on data sampled from the training set. While I expect you to design your AdaBoost program in such a way that you can plug in any weak learner, I would like you to use Decision Stumps this week. Decision Stumps are simply one-level decision trees. That is, the learner selects an attribute for the root of the tree and immediately classifies examples based on their values for that attribute. (So this assignment also gives you the opportunity to do a bit of simple decision tree implementation.)

In order to test your program, you can use the following datasets:

- `weather.nominal.arff`,
- `titanic.arff`, and
- `vote.noUnknowns.arff`

You will find these in the usual places, i.e., on Glow or in:

```
~andrea/shared/cs374/NominalData
```

These are binary classification problems and have data with nominal-valued attributes and no missing values. Because the `weather.nominal.arff` data set is small, perform a leave-one-out evaluation. For the other two, you can do a 10-fold cross validation.

As always, you can run weka's AdaBoost implementation on these data sets to give you a sense of the accuracy for which you should aim. (AdaBoost can be found under "meta" in weka when you're choosing a classifier.)