Assignment report, group number

Your Names Go Here

Leiden Institute of Advanced Computer Science, The Netherlands

Abstract. This document contains the format for the report required for submission of the practical assignment for the course Introduction to Machine Learning.

1 Introduction

This document serves as a *description of the first practical assignment* for the course Introduction to Machine Learning. For this assignment, you will implement an ensemble learning algorithm.

The purpose of this assignment is to gain familiarity with the inner workings of machine learning algorithms, translating the concepts from the lectures into working code. As such, you will be asked to implement a base learner (linear halfspaces). This learner can then be used as the basis for an ensemble learning algorithm using adaBoost. You will then investigate the performance of your algorithm on some artificial test-sets.

Since this assignment focuses on implementing the algorithms, you are **not** allowed to make use of exisiting machine learning libraries like sklearn. You are allowed and encouraged to make use of numpy for any mathematical operations you might need.

Next to this description, we provide a jupyter notebook which contains skeleton code you should use througout this assginment.

2 Base Learner

The base learner for this assignment will be linear halfspaces. You can make use of the batch perceptron algorithm, or implement (stochastic) gradient descent. The only requirement is that you make use of a learning rate parameter which controls the speed of the learning process.

Your code should be structured according to the provided skeleton code (so you implement the 'init', 'fit' and 'predict' methods). You can add more methods if needed, but don't modify the call signature of the provided methods. You should keep in mind the following:

- Implement an upper limit on the number of iterations, since we can't assume separability
- Make use of a learning rate which decreases over time
- Consider whether you prefer to start from $w = \vec{0}$ or from a randomly initialized w. What is the practical difference?

3 AdaBoost

Using your implementation of linear halfspaces algorithm as the base learner, you should implement an AdaBoost algorithm. This means you should do the following:

- Train your algorithm as the first weak learner
- Keep track of the probability distribution on the samples and update it every time you train a new perceptron. There are many ways in which you could adapt these samples, which you can compare when testing your algorithm in the next section.

4 Testing the algorithm

To test your algorithm, you can make use of the provided datasets. The skeleton code shows how to load these files. Your testing should be rigorous and documented clearly in your report. In particular, try to focus on the following questions:

- How does the ensemble compare to the base learner? Is the difference always present, why? Hint: there are 3 2D-datasets present in the testsuite, visualize your final models on both of these.
- How stable is your base algorithm to the choices of learning rate?
- How does the sample distribution update rule impact the performance of the ensamble? Hint: show how the performance of the ensamble changes in each iteration.

5 Extending to multiclass classification

You don't have to implement this, but you should consider how your final ensemble learner could be extended to the multiclass case. Write a section in your report discussing this.

6 Submission

For your submission, you should submit your report in the style of a scientific paper (introduction, background, experiments, results, conclusions...) of **maximum 8** pages (be concise). Make sure to clearly introduce (and cite) the methods you use (the base learning + the boosting algorithm).

You should also submit your code in the exact structure of the provided skeleton code, as **one** '.py'-file named `GroupXX_A1.py`.

References