The University of Adelaide, School of Computer Science

Introduction to Statistical Machine Learning

Semester 2, 2019 Assignment 2: Implementation of AdaBoost

DUE: 25 Sep. 2019, Wed 11:55 PM

Submission

Instructions and submission guidelines:

- You must sign an assessment declaration coversheet to submit with your assignment.
- Submit your assignment via the Canvas MyUni.

Reading

With this assignment, you will see how Adaboost works on a classification task. The AdaBoost algorithm is described in the class and more informtion on AdaBoost can be found on the web pages: https://en.wikipedia.org/wiki/AdaBoost

Please read "A Short Introduction to Boosting" by Yoav Freund and Robert E. Schapire, which can be found here: http://www.cs.princeton.edu/~schapire/papers/FreundSc99.ps.gz

and,

http://rob.schapire.net/papers/explaining-adaboost.pdf

If you find difficulties to understand this paper, you may read other tutorial/survey papers on the same webpage. If and only if you want to know more about Boosting methods, you are encouraged to read the following papers on Boosting (Optional):
https://arxiv.org/abs/0901.3590

https://digital.library.adelaide.edu.au/dspace/handle/2440/78929

https://arxiv.org/abs/1302.3283

Coding

You are provided with the training data (xi; yi); i = 1...., belonging to two classes, with binary labels yi (If yi is NOT {+1, -1}, you need to convert the labels into {+1, -1} first). You should use these training data to train an Adaboost classifier.

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 for this assignment. 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. Refer to: https://en.wikipedia.org/wiki/Decision_stump

To simplify the task, I have also provided a Matlab implementation of Decision Stump ("build_stump.m"). This is for reference only. Please be aware that you may need to rewrite/modify the decision stump code for your own needs.

There is a combinatorically large number of experiments that you could run and likewise, number of measures/settings that you can report against (training time, prediction on testing set, test time, number of boosting, depth of weak learners – your implementation only has to provide for Stumps but you can compare against Matlab/Python versions with deeper weak learners for Adaboost.

If you want, you can extend your code to have trees of some greater depth as weak learners). This assignment is deliberately open-ended and flexible, meaning that you can follow to some extent what interests you but also tests your ability to think strategically and work out what might be the most informative, interesting and efficient things that you could do (and report on).

Please be aware that there is the law of diminishing returns. Loosely put, you do a great job and you will get 9/10, and you do an amazing job and you will get 10/10. However, for the 10% extra marks you may well have done 400% more work.

Please start early. This might be a tough algorithm to implement and debug. You can choose either Matlab, Python, or C/C++ to implement AdaBoost. I would personally suggest Matlab or Python.

Your code should not rely on any 3rd-party toolbox. Only Matlab's built-in API's or Python/ C/C++'s standard libraries are allowed. When you submit your code, please report your algorithm's training/test error on the given datasets.

You are also required to submit a report (<10 pages in PDF format), which should have the following sections (report contributes 45% to the mark; code 55%):

- An algorithmic description of the AdaBoost method. (5%)

- Your understanding of AdaBoost (anything that you believe is relevant to this algorithm) (5%)

• Some analysis of your implementation. You should include the training/test error curve against the number of iterations on the provided data sets in this part (see above. This part is open-ended) (20% for master students and 25% for undergraduate students)

• You should compare performance with an “inbuild” package (such as fitemsemble in Matlab: https://au.mathworks.com/help/stats/fitensemble.html)   (5% for master students and 10% for undergraduate students)

• You may also train an SVM and compare the results of SVM with AdaBoost. What do you observe? (10% for master students. This task is optional for undergraduate students)

In summary, you need to submit (1) the code that implements AdaBoost and (2) a report in PDF.

Data

You will use Wisconsin Diagnostic Breast Cancer dataset to test your model. All the data points are stored in the file “wdbc_data.csv”. The explanation of the data field is given in “wdbc_names.txt”. You need to predict diagnosis of each sample based on the real-valued features.

There are 569 samples in “wdbc_data.csv”. You will use the first 300 samples for training and use the remaining part for testing.