Assginment 1

April 6, 2016

Due: April 10, 2016, 11:59 PM PDT
Late policy: Every 10% of the total points will be deducted for every extra day past due.

1 Introduction

In multi-class classification, each training data point belongs to one of the N different classes. The goal is to construct a function that given a new data point, will correctly predict the class to which the new data point belongs. In this assignment, you will explore the solutions to the multi-class classfication tasks and compare with the one vs. all strategy and specific multi-class classification algorithms.

2 Classifier

You will choose at least ONE classifiers to perform the experiments, common choices include, but not limited to:

- **SVM**

  For the SVM classifier, you can use the libsvm
  `http://www.csie.ntu.edu.tw/cjlin/libsvm/`
  or the svm-light
  `http://download.joachims.org/svm_multiclass/current/svm_multiclass.tar.gz`
  Descriptions of the usage of (multiclass) SVM can be found at:
  `http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html`
  and `http://svmlight.joachims.org/svm_multiclass.html`

Another nice package which solves SVM in the primal space is Pegasos:
`http://www.cs.huji.ac.il/~shais/code/pegasos.tgz`.
The Pegasos package is focused on the two-class classification problem. You will need to implement your own version of Pegasos for the multi-class case, but with the same spirit as in the two-class case.

- **Boosting**

  You can use/implement the AdaBoost algorithm and apply it on one vs. all strategy and implement multi-class AdaBoost algorithm, For example, AdaBoost.MH algorithm [1]. See [4] for a review. Some existing libaries include multiboost (`http://www.multiboost.org/download`) and sciki-learn (`http://scikit-learn.org/stable/auto_examples/ensemble/plot_adaboost_multiclass.html`)

- **Random Forest**

  Using random forest classifier is another option.
  You can train K, the total number of classes, 2-class random forests classifiers and train a directly random forest to deal with the multclass classification. `http://www.stat.berkeley.edu/~breiman/RandomForests/`
  In this case, you don't need to change anything in the random forest classifier code. All you need to do is to write a wrapper to do that.

3 Dataset

You will choost at least ONE dataset to perform the study. Though the choice of datasets, is also flexible, we encourage you to use standard machine learning datasets from UCI repository. A subset of datasets, focusing on multi-class classification tasks, can be downloaded from:

`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`

You can use other datasets from the UCI repository for your own research.

4 Requirement

Train and test classifiers: one vs. all and explicit multi-class classifier on benchmark datasets.

Write a report including: a) abstract, b) method, c) experiment, d) discussion, and e) references. You can follow leading conferences like NIPS (`https://papers.nips.cc/`) or ICML (`http://icml.cc/2016/?page_id=151`).

Having a thorough conclusion is always a plus if you can show the difference w.r.t. the number of classes, varying amount of training and testing data, and specific family of models.

Copy and paste your own code as appendix section to your report.

We strongly encourage you to read and learn how to proceed your study through the following two highly cited review/analysis papers:

> A Comparison of Methods for Multiclass Support Vector Machines, Chih-Wei Hsu and Chih-Jen Lin [2]
>
> In Defense of One-vs-all Classification, Ryan Rifkin and Aldebaro Klautau [3]

5 Grading

Your grade will be based on

- How challenging and large are the datasets you are studying?
- Any aspects that are new in terms of algorithm development, uniqueness of the data, or new applications?
- Is your experimental design comprehensive? Have you done thoroughly experiments?
- Is your report written in a professional way?
- Bonus will be given if you can use multiple classifiers and test on multiple datasets.
- As always, there will be bonus applied if you can implement your own algorithm from scratch, instead of using off-the-shelf libraries.
- Besides One vs. All and explicit multi-class classifiers, we also mentioned the (Error-Correcting) Output-Code multi-class strategy. Bonus will be given if you would implement the output-code strategy for a multi-class classification problem with a large number of classes.

References

[1] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[2] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[3] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

[4] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost.