Middle East Technical University   Department of Computer Engineering

CENG 499

Introduction to Machine Learning

Fall 2015-2016

Term Project

Boosted Classification

Due date: Jan 25, 2015, Sun, 23:55

1 Introduction

Problem of interest in the term project is identifying patients with Parkinson's disease using features extracted from sound recordings of patients and non-patients exercising informative speaking tasks under medical surveillance. Associated study is conducted by Istanbul University, Bahcesehir University and Bogazici University in joint and outcome of this study is elaborated in the paper with title "Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings" published on *IEEE Journal of Biomedical and Health Informatics* in 2013. Please read this article and summarize it shortly in your term paper, in which you will also compare your obtained results with theirs.

You are going to devise, implement and evaluate a **boosted classifier ensemble meta-algorithm** that combines the outputs of **weak** learners such as **k-nearest neighbors**, **Naïve Bayes** and **discriminant analysis** procedures to address this **binary** classification problem on MATLAB framework.

Training data for this task is included in the file named "*train_data.prj*" dated back to 2014. Following lines contain the 1040 data instances each comprising of 26 attributes followed by the class label whether the associated health condition is present or not. Beware that file contains 28 columns before the label, yet information on the first and last columns should not be considered as extracted features as they are identification numbers. Data set is publicly available on UCI Machine Learning Repository, under this link to which you may refer to regarding raw sound files of patients and other details.

You are also provided with a separate test file with identifier "*test_data.prj*" holding 168 instances. Your boosted algorithm will be evaluated on this data set and accuracy you obtain upon this action will indicate the performance of your algorithm.

This file is continued with a very quick review of constructing ensemble classifiers using the adaptive boosting meta-learner and the steps you should take concerning your implementation of the task, as organized in the next section.

2 Adaptive Boosting

Assume that you are given the training data set $T = \{(\mathbf{x}_i, y_i) \,|\, i \in [1..m]\}$, in which each $\mathbf{x}_i$ is an n-dimensional multivariate vector, and each $y_i \in \{0, 1\}$ denotes the class label, where 1 indicates that corresponding $\mathbf{x}_i$ is an instance pertaining to some subject having Parkinson's disease, while 0 indicates that the associated subject does not exhibit the medical condition.

A weak classifier can be reflected as a classification method whose prediction accuracy on the validation set is greater than 0.5, so it is better to use this method than random guessing for the binary classification problem. Your K weak learners are represented by the functions they approximate during training as $h_{\theta_k}(\mathbf{x})$ indexed by $k \in [1..K]$, whose output is the predicted class label for a multidimensional vector $\mathbf{x}$.

Outputs of weak classifiers are to be combined using the Adaptive Boosting (AdaBoost) meta-learner outlined in the following.

Algorithm 1 Adaptive Boosting Algorithm

```
1: procedure ADABOOST($\mathbf{X}, \dot{\mathbf{x}}, K$)
2:     Initialize observation weights, $w_i = 1/m$, $i = 1, 2, ..., m$
3:     $k \leftarrow 1$
4:     while $k \neq K$ do
5:         Fit classifier $h_{\theta_k}(\mathbf{x})$ to the training set using weights $w_i$
6:         Compute error
               $$err^{(k)} \leftarrow \frac{\sum_{i=1}^{m} w_i \, \mathbb{1}\big(y_i \neq h_{\theta_k}(\mathbf{x}_i)\big)}{\sum_{i=1}^{m} w_i}$$
7:         Compute trust
               $$\alpha^{(k)} \leftarrow \log \frac{1 - err^{(k)}}{err^{(k)}}$$
8:         Update weights
               $$w_i \leftarrow w_i \exp\left(\alpha^{(k)} \, \mathbb{1}(y_i \neq h_{\theta_k}(\mathbf{x}_i))\right), \; i = 1, 2, ..., m$$
9:         Normalize weights
               $$w_i \leftarrow \frac{w_i}{\sum_{i=1}^{m} w_i}, \; i = 1, 2, ..., m$$
10:        $k \leftarrow k + 1$
11:    end while
12:    return $C(\dot{\mathbf{x}}) = \arg\max_j \sum_{k=1}^{K} \alpha^{(k)} \, \mathbb{1}\big(h_{\theta_k}(\mathbf{x}_i) = j\big)$
13: end procedure
```

AdaBoost represented in Algorithm 1 trains the week classifiers iteratively by altering weights of observations in the training set. Initially, all data instances get equal weights. In other words, they are equally important. Yet, over time persistently misclassified instances acquire greater weights so that we may put more trust in classifiers that correctly classify them.

Consequently, algorithms executing correct predictions over more important data instances will get more degrees of trust. Note that if a classifier has more than half error rate, negative trust points will be assigned to them.

Determination of class label of a data vector is based on a voting scheme in which vote of an algorithm is worth basically the degree of trust it obtained in the training procedure.

3 Programming and Documentation Tasks

Change the default format of the MATLAB workspace into long fixed-decimal format to avoid possible numerical errors.

Load both training data into your choice of data structures. An example would be dictating the attribute-related information to be stored on data design matrix $\mathbf{X}$ and corresponding class-labels to be stored on some vector $\mathbf{y}$. Load testing data as well.

Perform **feature scaling**. Write down the scaling method you used.

You will **not** implement the weak classifiers. Use MATLAB Statistics and Machine Learning Toolbox functions **fitcknn**, **fitcnb**, and **fitcdiscr** for k-nearest neighbors, Naïve Bayes and discriminant analysis classifiers, respectively. Similarly, use MATLAB **predict** functions to obtain predicted class labels. You may study these algorithms using well-documented information on associated MATLAB pages.

You are going to implement **two** versions of AdaBoost algorithm. In the first version, you are going to train each classifier with the whole data set. On the other hand, in the second version, you will select a portion of training data set using weights as probabilities that corresponding samples may reside in the selected set. Maximum size of the selected set at some iteration being less than the size of the whole training set is a parameter and should be **varied**. Do **not** use any readily-uploaded MATLAB code on repositories for this task.

For all experiments you are going to plot **Receiver Operating Characteristic** curves, and also you may use an evaluation metric you decide to utilize via research, together with the regular prediction accuracy we have computed up to this point. You may employ built-in MATLAB commands regarding that. Give citations to used codes.

Ensuring that each algorithm is used in at least **ten** different settings under the constraint that none of these settings incur specification of supervised cost/risk information, run your first version AdaBoost implementation and calculate boosted prediction accuracy on the training set and accuracy on the testing set. Compare the results you obtained with the best accuracy reading of a single method among weak classifiers. How did the accuracy change?

Varying the subsample size parameter, conduct the same experiment this time using your second version AdaBoost implementation. Document your findings.

Visualize AdaBoost classification on both training and testing data set with the highest prediction accuracy by projecting the related data onto the **first two principal components**. Mark incorrect classifications.

Perform **feature selection** using Principal Components Analysis. Select top-P principal components that explain more than at least 80% of the total variance. Conduct same experiments using these new features. Discuss and visualize your findings.

Term project report, at least 4 and at most 8 pages long in IEEE format, should contain theoretical and experimental issues and all of your findings with reference to the article of this study and other resources. Report should reflect your methodology clearly and also the discussion you would like to bring up into the matter. Also reflect any possible future studies for the problem in question. Documentation is clearly very important for ML projects, but in this case you should be paying more attention to the writing process than you did during the homeworks. It should include tables and figures you have generated in the task with your comments. Try to follow an outline similar to the original article of the study, which exemplifies ML methodology on the matter clearly.

BONUS: Top-two students who obtain the highest prediction accuracies on the testing set will receive 10% and 5% bonus upon their total grades, respectively.

ADDITIONAL BONUS: Using another dimensionality reduction technique such as Fisher Discriminant Analysis or Kernel Principal Component Analysis, or some state-of-the-art feature selection method, conduct the same experiment once more. If overall accuracy is increased, you will get an additional 10% bonus upon your total grade. You may use available MATLAB implementations with citation.

4 Restrictions and Tips

- Do not use any available MATLAB repository files without referring to them in your report.
- Vectorization is essential regarding the performance.
- Do not use MATLAB toolbox functions or readily uploaded codes for Adaptive Boosting algorithms.
- Implementation should be of your own in a sense that it reflects your thinking of the ML methodology. Readily-used codes should not exceed a reasonable threshold within your total work.
- Do not forget that the code you are going to submit will also be subject to manual inspection.

5 Submission

- **Late Submission:** No late submission is accepted.
- Your scripts and function files together with a 4-to-8 pages long report focusing on theoretical and practical aspects you observed regarding this task should be uploaded on COW before the specified deadline as a compressed archive file whose name should be <<**student_id**>_**project**> preceding the file extension. Write down how your code is supposed to compile in some README.
- The archive must contain **no directories** on top of MATLAB/Octave scripts and function files.

6 Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.
2. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.