Adaboost Algorithm with Different Versions

Samet Aytaç
Middle East Technical University/Computer Engineering Department
samet.aytac@ceng.metu.edu.tr

***Abstract*—This document tries to show how can we increase Adaboost algorithm's efficiency with some modification.**

***Keywords*—adaboost, boosting, PCA, probabilisticPCA, feature selection, weak classifier**

I. INTRODUCTION

Problem of interest in the paper 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. In this paper, we are going to show how variation of Adaboost algorithm perform on that particular dataset.

II. TECHNICAL EXPLANATIONS OF ALGORITHMS

For this report, I scale both training and test data. I divide every column with its max element, so every elements scale in range (0, 1).

For adaboost, I use

Initialization...
For $t = 1, ..., T$:

- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1+r_t}{1-r_t})$
- Update

$$D_{t+1}(i) = \frac{D_t(i)exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Output the final classifier:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

First version of adaboost is directly implementation of this algorithm. I select 30 weak classifiers which are variation of k-nearest neighbors, Naive Bayes and discriminant analysis algorithms.

Second version of adaboost is a bit different. For second version, I select a portion of training data set using weights as probabilities that corresponding samples may reside in the selected set. I use datasample() function for select data which is built-in function of Matlab.

Then, I apply feature selection operation to dataset by Principal Components Analysis. I use pca() function for feature selection which is built-in function of Matlab. I select top-P principal components that explain more than at least 80% of the total variance.

Finally, I apply feature selection operation to dataset by probabilistic Principal Components Analysis. I use ppca() function for feature selection which is built-in function of Matlab.

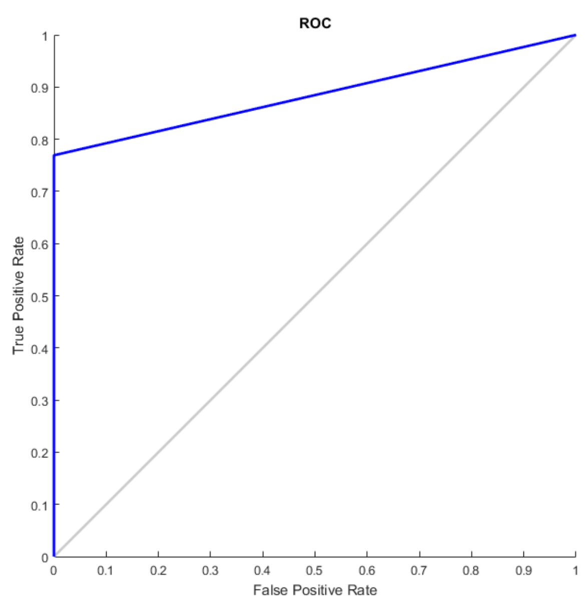
III. RESULTS OF EXPERIMENTS

In this chapter, I will explain what experiments perform on dataset and briefly conclude them.

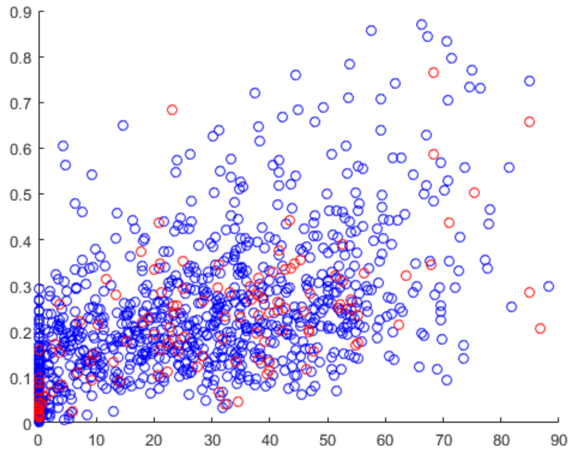
A. Experiment 1

In this experiment, I use adaboostV1. I create model with training data and I make prediction on training data. I find accuracy=0.88. Most of the weak classifiers (e.g. discriminant analysis) perform accuracy=1. It is normal because dataset is not very complicate.

Here is Receiver Operation Characteristic curve:



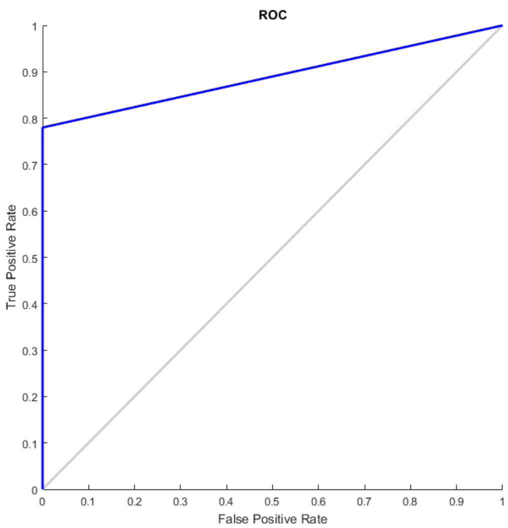
Here is visualization of the data. Selected two features are the first two principal components. Wrong guesses are marked as red circle.



B. Experiment 2

In this experiment, I use adaboostV1. I create model with training data and I make prediction on test data. I find accuracy=0.77.Discriminant analysis performs accuracy=0.99.Since test data has only 168 data points, it is normal. If dataset would be much bigger, adaboost will perform much better than Discriminant analysis. Another point is, remember that we find 0.88 accuracy in experiment1. So we can say there is an over fitting.

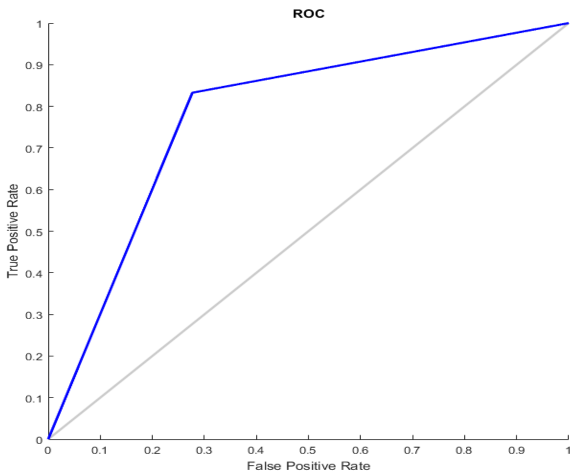
Here is Receiver Operation Characteristic curve:



C. Experiment 3

In this experiment, I use adaboostV2. I create model with training data and I make prediction on training data. I choose maximum subspace=500 data points. I find accuracy=0.74. It performs worse than adaboostV2 for training data. Actually, it makes sense because in every iteration, it takes some subspace of training data, not all of them.

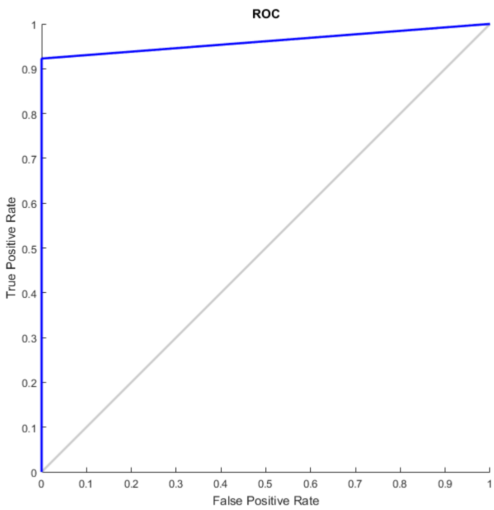
Here is Receiver Operation Characteristic curve:



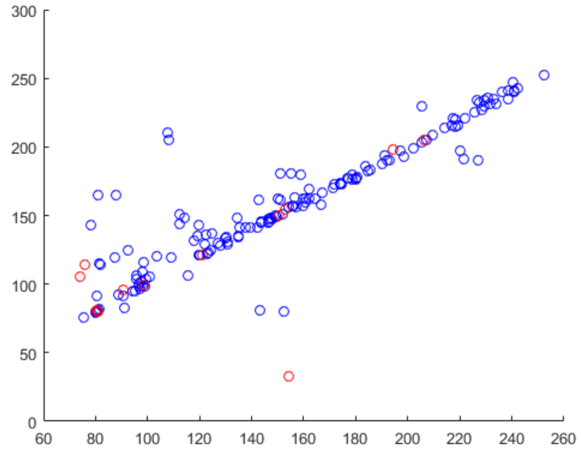
D. Experiment 4

In this experiment, I use adaboostV2. I create model with training data and I make prediction on test data. I choose maximum subspace=500 data points. I find accuracy=0.91. So, it performs much better than experience2. We can figure out that, by using AdaboostV2, we can make more accurate predictions. Moreover, remember that we have 0.74 accuracy in experiment3. So, we solve over fitting problem by using AdaboostV2.

Here is Receiver Operation Characteristic curve:



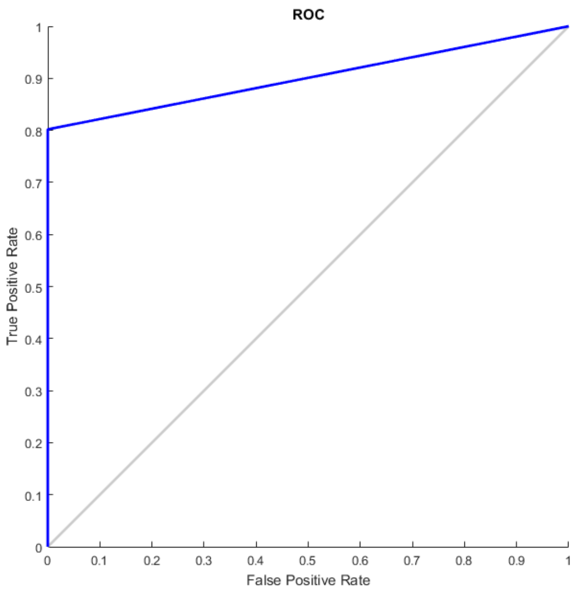
Here is visualization of the data. Selected two features are the first two principal components. Wrong guesses are marked as red circle.



E. Experiment 5

In this experiment, I use adaboostV2. I create model with training data and I make prediction on training data. I choose maximum subspace=700 data points. I find accuracy=0.88. So, remember in experiment 3 which we use subspace=500, accuracy was 0.74 . When number of subspace data points increase, accuracy for training set increase. It makes sense because now our subspace is more close to training set.

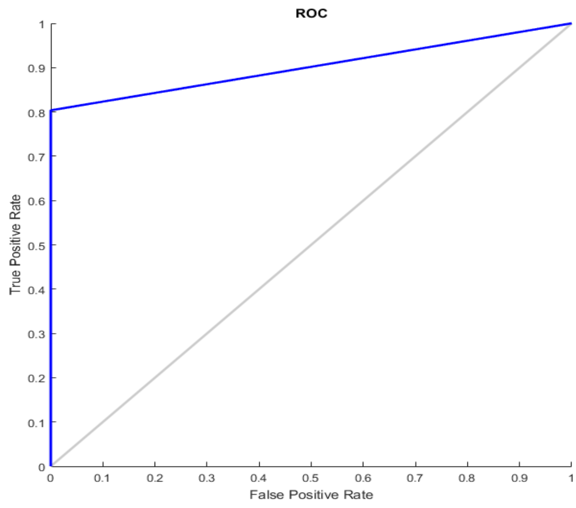
Here is Receiver Operation Characteristic curve:



F. Experiment 6

In this experiment, I use adaboostV2. I create model with training data and I make prediction on test data. I choose maximum subspace=700 data points. I find accuracy=0.83. Remember, we find 0.88 accuracy in experiment 5. It means, when we increase size of subspace, it is more likely to suffer from over fitting.

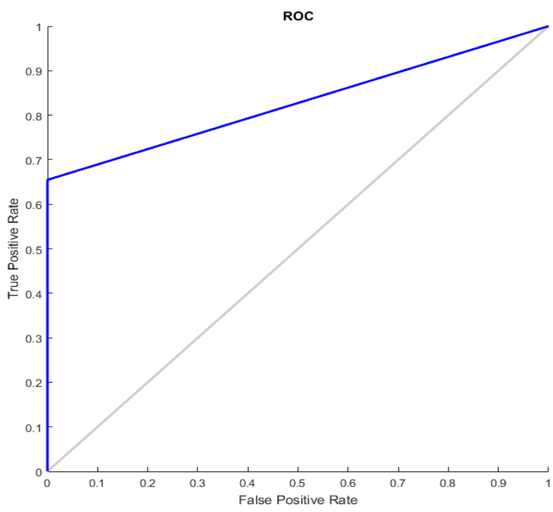
Here is Receiver Operation Characteristic curve:



G. Experiment 7

In this experiment, I use adaboostV1 with PCA. I create model with training data and I make prediction on test data. I choose features top-P principal components that explain more than at least 80% of the total variance. I find accuracy=0.65

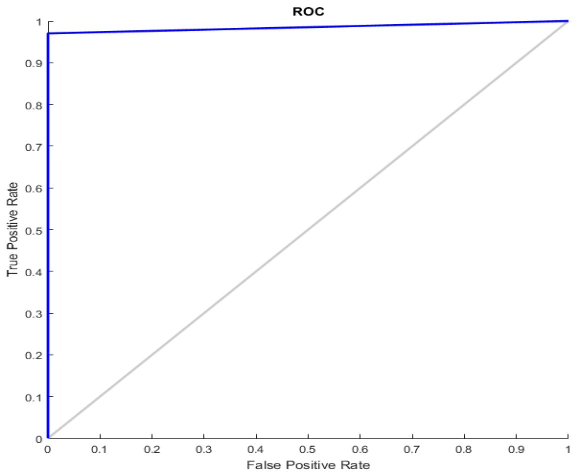
Here is Receiver Operation Characteristic curve:



H. Experiment 8

In this experiment, I use adaboostV2 with PCA. I create model with training data and I make prediction on test data. I choose maximum subspace=500 data points. I choose features top-P principal components that explain more than at least 80% of the total variance. I find accuracy=0.96. Remember that without PCA, accuracy of adaboostV2 is 0.91(experiment4). So PCA is increase accuracy for adaboostV2.

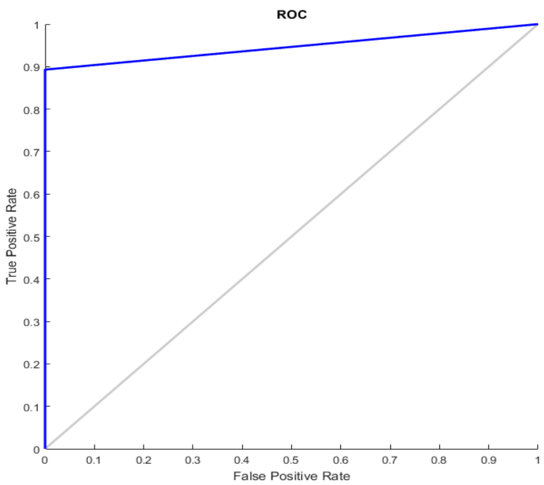
Here is Receiver Operation Characteristic curve:



I. Experiment 9

In this experiment, I use adaboostV1 with PPCA. I create model with training data and I make prediction on test data. I choose features top-10 principal components. I find accuracy=0.89. Remember that without PPCA, accuracy of adaboostV1 is 0.77(experiment2). So PPCA is increase accuracy for adaboostV1.

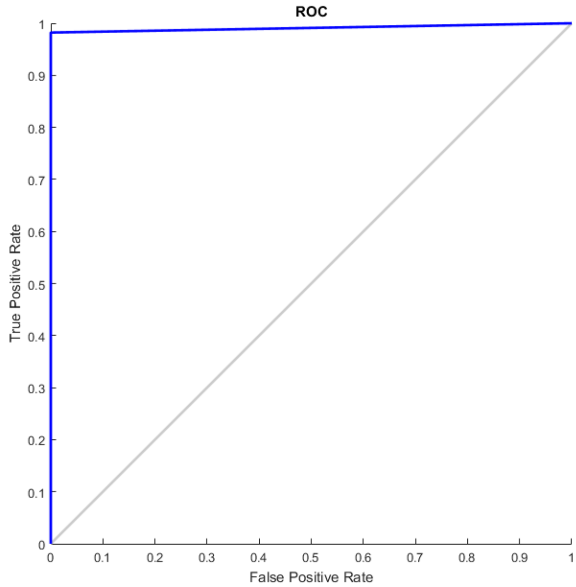
Here is Receiver Operation Characteristic curve:



J. Experiment 10

In this experiment, I use adaboostV2 with PPCA. I create model with training data and I make prediction on test data. I choose features top-10 principal components. I choose maximum subspace=500 data points. I find accuracy=0.98. Remember that without PPCA, accuracy of adaboostV1 is 0.91(experiment4). So PPCA is increase accuracy for adaboostV2.

Here is Receiver Operation Characteristic curve:



IV. Conclusion

To conclude our experiments, we can say that adaboostV1 is suffers from over fitting. About adaboostV2, it is very critical to find optimum size of subspace. If we choose subspace too small, our prediction will suffer from under fitting. If we choose subspace too large, our prediction will suffer from over fitting. About dimensionality reduction techniques, they are increase accuracy. For this dataset, the best way is using adaboostV2 with PCA or PPCA.

References

[1] Sakar, B., Isenkul, M., Sakar, C., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H. and Kursun, O. (2013). Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4), pp.828-834.

[2] Mathworks.com, (2016). *k-nearest neighbor classification - MATLAB*. [online]Available at: http://www.mathworks.com/help/stats/classificationknn-class.html [Accessed 25 Jan. 2016].

[3] Mathworks.com, (2016). *Train multiclass naive Bayes model - MATLAB fitcnb*. [online] Available at: http://www.mathworks.com/help/stats/fitcnb.html [Accessed 25 Jan. 2016].

[4] Mathworks.com, (2016). *Fit discriminant analysis classifier - MATLAB fitcdiscr*.[online] Available at: http://www.mathworks.com/help/stats/fitcdiscr.html [Accessed 25 Jan.]