Boosting Weighted Linear Discriminant Analysis

Kazunori Okada[1], Arturo Flores[2], Marius George Linguraru[3]

[1]Computer Science Department, San Francisco State University,1600 Holloway Avenue, San Francisco, CA 94132, USA, kazokada@sfsu.edu

[2]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA

[3]Radiology and Imaging Sciences, Clinical Center, National Institutes of Heath, 10 Center Drive, Bethesda, MD 20892, USA, lingurarum@mail.nih.gov

Abstract

We propose a novel approach to boosting weighted linear discriminant analysis (LDA) as a weak classifier. Combining Adaboost with LDA allows to select the most relevant features for classification at each boosting iteration, thus benefiting from feature correlation. The advantages of this approach include the use of a smaller number of weak learners to achieve a low error rate, improved classification performance due to the robustness and stable nature of LDA, and computational efficiency. The performance of the proposed method was evaluated on artificial data and additionally on two popular independent data sets: the Iris Data Set and the Breast Cancer Wisconsin Diagnostic Data Set, both publicly available at the University of California at Irvine Machine Learning Repository. Experimental results showed the superior accuracy of the proposed method over LDA and AdaBoost combined with other types of weak classifiers. The weighted LDA algorithm was proven to be equivalent to the traditional LDA in the case of uniform weight distributions.

Keywords: Classification, Adaboost, weighted linear discriminant analysis..

1. Introduction

Adaboost [5] is one of the most popular meta/ensemble learning algorithms that build an accurate (strong) classifier from a group of inaccurate (weak) classifiers. In its classic formulation, the weak classifier of Adaboost is defined by a simple function that performs basic binary thresholding on a single feature extracted from each training sample. Hence, each weak classifier is trained to minimize a weighted error on a single feature. Positive-valued weights are introduced to each training sample and adapted during the course of training in order to focus on samples that are difficult to classify. At each iteration, Adaboost selects the feature that minimizes the weighted error. In this respect, Adaboost is essentially a feature selection algorithm. Finally, the strong classifier of Adaboost combines in a linear fashion multiple weak classifiers selected according to the sample weights, resulting in a joint single decision rule. Adaboost has been successfully applied to various medical and non-

medical applications such as lung tumor detection [1], Alzheimer's disease detection [10] and pedestrian detection [15] to name a few.

Alternatively, Linear Discriminant Analysis (LDA) [2] is one of the classic statistical pattern classification techniques. The goal in LDA is to find an optimal subspace from a feature vector space, which maximizes the ratio of 'between-class scatter' to 'within-class scatter' [2]. As a result, LDA benefits from feature combinations that produce the highest separation between classes.

This paper presents a new approach to combine the Adaboost and LDA algorithms to improve classification performance by exploiting the LDA classifiers as weak classifiers of Adaboost. By combining these two algorithms, Adaboost can select the best feature combination at each boosting iteration instead of a single feature, therefore taking advantage of feature correlations. The classic LDA theory, however, does not include sample weights that are necessary for the boosting framework. In order to adopt LDA as weak learner of Adaboost, we introduce sample weights in the LDA formulation.

Several previous studies relate to our work. An outlier-class resistant approach to estimate the within-class covariance matrix in LDA for linear dimensionality reduction was presented by Tang et al. [13]. Several methods were proposed to estimate the LDA weights and tests were performed on medical and non-medical data. Although this work explored similar sample weights in their LDA formulation, they did not investigate its application in the boosting context of our interest. The adaptation of other classification models to Adaboost has recently been explored [8,14], however there are only very few reports that investigated LDA in this context, despite the popular usage of discriminant analysis [9,11]. An interesting approach was proposed by Liu et al. [9] who adapted LDA to incorporate sample weights in an application related to indoor/outdoor digital image classification. While the authors introduced a weighted LDA - Adaboost combination, their approach was not equivalent to a traditional LDA in the case of uniform weight distributions. As AdaBoost initializes the sample weights to a uniform distribution, it is important that a true LDA classifier be applied in the first iteration. This condition can be seen as a guarantee that the initial classifier represents an adequate starting point. Skurichina and Duin [11] also investigated boosting LDA classifiers; however their scheme did not aim to combine the two algorithms by introducing weights in LDA formulation.

To evaluate the effectiveness of the proposed approach, we apply the method first to artificial data and then to two independent data sets and compare our method with AdaBoost combined with other types of weak classifiers. The experimental results reflect the superior accuracy of the proposed method over LDA and AdaBoost.

2. Data and Methods

The following sections describe the proposed method for boosting weighted LDA classifiers. First the experimental data are presented followed by some basic methodological concepts used in this study.

2.1 Data

For the evaluation of our method, we use two publicly available data sets, the Iris Data Set [3] and Breast Cancer Wisconsin Diagnostic Data Set (BCWDDS) [12], both from the University of California Irvine Machine Learning Repository [17]. The iris data contain 150 instances with 4 attributes, which are grouped in 3 classes corresponding to a type of iris plant. One class is linearly separable from the other

two; the latter are not linearly separable from each other; we employed the 100 cases that are not linearly separable in our experiments. More precisely, a pair of labels for versicolor and virginica are used for our binary classification task. The BCWDDS data are 569 instanced from benign (357) and malignant (212) cases each with 32 attributes. Features were computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image.

Typically, the raw input to a classification system is first subjected to a feature extraction stage in order to transform the information into a more discriminant representation of the data. Our study employs the entire feature space of the two data sets: 4 features for the iris data [3] and 32 features for BCWDDS [12].

Additionally, a two dimensional artificial/simulated data set was used to validate the algorithm in a very low feature space (Figure 1). The artificial data set consists of 500 samples evenly divided between two classes. Samples from one class are clustered on a central point and are completely surrounded by samples of the second class. The data of the two classes are clearly not linearly separable.

2.2 Adaboost

Adaboost [5] is an ensemble learner where the joint decision rule of multiple weak classifiers forms an overall strong classifier. Adaboost assigns an initial uniform weight $W_1(i)=1/n$ to each training sample x_i, where n is the number of training samples. At iteration k, Adaboost finds the classifier h_k trained using samples x_i that minimizes the weighted error E_k according to $W_k(i)$. The weights are updated by

$$W_{k+1}(i) = \frac{W_k(i) \cdot \exp(-\alpha_k y_i h_k(x_i))}{Z_k}, \tag{1}$$

where $\alpha_k = \frac{1}{2}\log\left(\frac{1-E_k}{E_k}\right)$, y_i are the ground truth labels, and Z_k is a normalization factor. According to these formulae, weights for correctly classified samples are increased and weights for incorrectly classified samples are decreased. This allows Adaboost to focus on the informative and difficult training samples. The resulting classifier of the Adaboost algorithm becomes

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right), \tag{2}$$

where h_t is the t-th weak classifier hypothesis, and $H(x)$ is the strong classifier hypothesis. Adaboost is typically combined with a simple threshold classifier that minimizes the weighted error on a single feature. In other words, at iteration k, Adaboost selects the feature which minimizes the weighted error. $W_i(k+1)$ is increased when x_i is classified correctly by h_k, and decreased otherwise. In this respect, Adaboost is essentially a feature selection algorithm.

2.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [2] is designed to maximize the ratio of 'between-class scatter' to 'within-class scatter'. Between-class scatter measures the

variance of the projections for each individual class. Within-class measures the variance of the projections from all data samples.

LDA employs feature combinations that produce the highest separation between classes. The traditional LDA implementation does not take into consideration weights. Supposing that we are given *N* training samples that are partitioned into *K* classes, $S_1,\ldots,S_c,\ldots,S_K$. Then the traditional LDA formulae for between-class scatter S_B and within-class scatter S_W are

$$S_B = \sum_{c=1}^{K} N_c \left(\mu_c - \bar{x}\right)\left(\mu_c - \bar{x}\right) \tag{3}$$

$$S_W = \sum_{c=1}^{K} \sum_{i \in S_c} \left(x_i - \mu_c\right)\left(x_i - \mu_c\right)^T \tag{4}$$

$$\mu_c = \frac{1}{N_c} \sum_{i \in S_c} x_i \tag{5}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i = \frac{1}{N} \sum_{c=1}^{K} N_c \mu_c \,, \tag{6}$$

where N_c is the number of cases in class *c*. The LDA criterion function can be written as

$$J(w) = \frac{w^T S_B w}{w^T S_w w}. \tag{7}$$

LDA finds the vector *w* where the *J*(*w*) is maximized. This vector is used to apply a linear transformation to the data, i.e. projecting the data onto the vector *w* by $y=w^T x$. After this transformation, the separation between classes is maximized. In order to have a complete classifier, a threshold w_0 is necessary. For a binary classifier, a typical choice of threshold is to compute the projections of the class means, μ_c, and then compute the mean of these two projections. A training sample can now be classified as positive if $w^T x_i > w_0$, and negative otherwise.

2.4 Weighted LDA

In order to use LDA with Adaboost, the LDA formulae are modified to take into account weights. By combining Adaboost with LDA, Adaboost can select the best feature combination at each boosting iteration instead of a single feature. We propose the reformulation of the scatter matrices for LDA with sample weights as follows

$$S'_B = \sum_{c=1}^{K} \left(\sum_{i \in S_c} P(i) \right) \left(\mu'_c - \bar{x'}\right)\left(\mu'_c - \bar{x'}\right)^T \tag{8}$$

$$S'_W = \sum_{c=1}^{K} \sum_{i \in S_c} P(i) \left(x_i - \mu'_c\right)\left(x_i - \mu'_c\right)^T \tag{9}$$

$$\mu'_c = \frac{1}{\sum_{i \in S_c} P(i)} \sum_{i \in S_c} P(i) x_i \tag{10}$$

$$\overline{x'} = \sum_{i \in S_c} P(i) x_i \,, \tag{11}$$

where $P(i)$ is the weight assigned to each i-th training sample with $\sum_{i=1}^{N} P(i) = 1$, as in the Adaboost algorithm. Instead of selecting the single feature that minimizes the classification error, with weighted LDA (wLDA) the best feature combination of features can be selected at each iteration, therefore taking advantage of feature correlation.

In the case of uniform weight distribution, namely $P(i) = 1/N$, the wLDA formulae become equivalent to the traditional LDA formulae. Another related formulation was proposed by Liu et al [9], but it did not incorporate sample weights into S_B and S_W. Hence, the formulation in [9] does not guarantee the equivalence to the traditional LDA, which can lead to inconsistent results, especially in a low dimensional feature space.

2.5 Classification via Adaboost with Weighted LDA

Recall that the traditional Adaboost implementation searches across simple threshold classifiers trained on a single feature. To combine Adaboost with wLDA, we train a wLDA classifier with all possible combinations of two and three features at each iteration. The feature combination that minimizes the weighted error is then selected. The sample weights are updated at each iteration according to equation (1). Note that the threshold w_0 also incorporates the weights, since it is computed using the projection of the weighted class means μ'_c.

2.6 Validation

A set of features were available for each instance in a data set. These features and ground truth labels were used to train the classifiers. We compare our proposed classification method combining Adaboost with the weighted LDA (AB+wLDA) against two related boosting methods: the original Adaboost with simple threshold weak classifiers (AB+ST) and another adaptation of Adaboost with weighted LDA (AB+wLDA_Liu) proposed by Liu [9]. AB+ST uses the weak classifier based on the simple threshold that minimizes the weighted error on a single feature. AB+wLDA_Liu offers a solution for boosting wLDA that does not incorporate sample weights in S_B, resulting in unbalanced scatter matrices that do not converge to the original LDA when using the uniform weights. Several classification performance metrics were used including test/training error at each boosting iteration and Leave K Out Cross Validation (LKOCV) [7]. For the iris and BCWDDS data, the LKOCV fold size was $k = 30$ and 210, respectively, and averaged over four test.

Initial tests are performed on the artificial data set to compare the ability of the discussed classifiers to distinguish instances from data with a low feature space. Results on the iris and BCWDDS data compare the same classifiers on instances with higher feature space.

3. Results

3.1 Artificial Data

A comparative example of classification from artificial data can be seen in Figure 1. The simple threshold classifier can only use one feature at a time. When Adaboost is combined with this type of weak learner (AB+ST), Adaboost selects the single feature that minimizes the weighted error at each iteration, which can be seen as the vertical and horizontal lines in Figure 1. On the other hand, LDA uses feature combinations instead of single features, thus taking advantage of feature correlations to separate between classes.

In both training and test error (Figure 1), the results for AB+ST and AB+wLDA are similar. A high number of similar classification rules can be seen in a localized area for AB+wLDA_Liu. In this low feature space AB+wLDA_Liu is equivalent to random guessing.

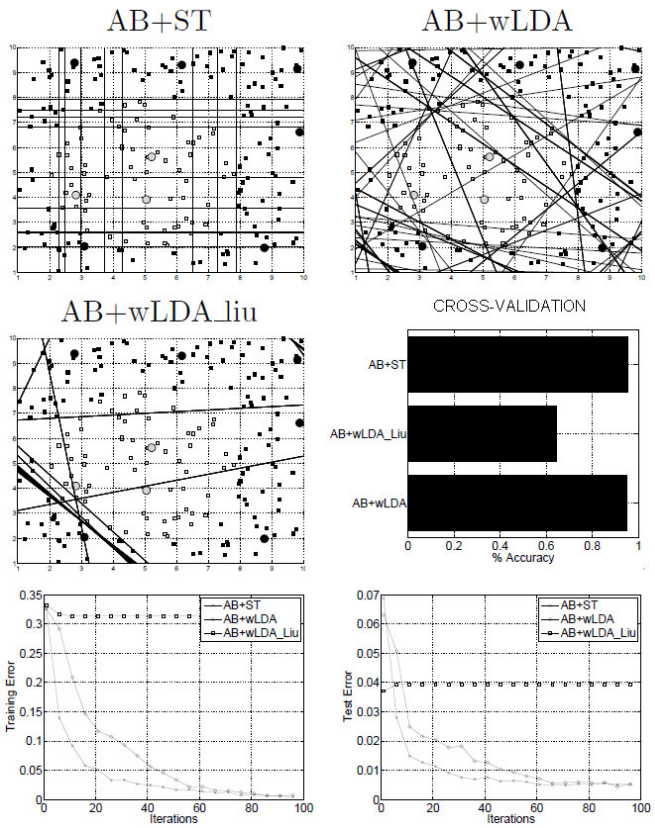

Figure 1. Comparison of classification techniques on artificial data. The top two rows show the classification results of AB+ST, AB+wLDA, and AB+wLDA_Liu along the average accuracy (cross-validation) of the techniques. The bottom row presents training and test errors for each trial run. Light colors in the data represent one class centered on a point; dark colors embody the second class surrounding the first class. Squares represent training samples and circles show test samples. Each line represents the decision boundary of a learned weak classifier. The cross-validation results for AB+wLDA_Liu show that in this low feature space, the method is equivalent to random guessing.

3.2 Iris Data Set

The wLDA implementation was further tested on the two non-linearly separable classes of the iris data set and results are presented in Figure 2. The iris data have a higher feature space (size 4), and we tested AB+wLDA by searching for the best three feature combination. AB+wLDA outperforms the other two classifiers (AB+ST and AB+wLDA_Liu) in the cross-validation shown in the bottom of Figure 2. In particular, AB+wLDA has a clear advantage over AB+ST on the higher feature space.

With AB+ST, the test error increases slightly as the number of boosting iterations increases, which indicates that AB+ST is over-fitting the data. With AB+wLDA there is a slight initial increase in the test error, but then it drops down the starting test error at the first iteration and remains constantly low. The strength and robustness of Adaboost combined with our wLDA implementation are also noticeable in the training set.

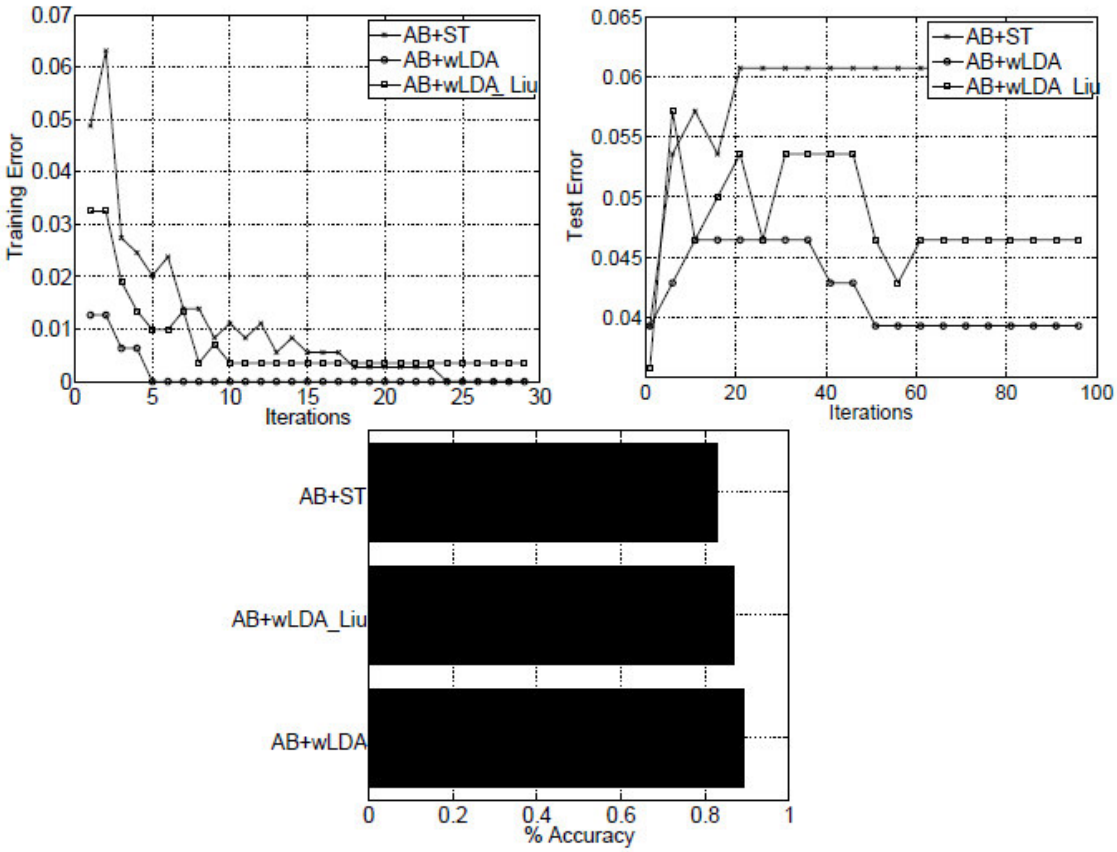

Figure 2. Experimental results on the iris data. The top row presents training and test errors for each trial run of LKOCV, while the bottom row shows the average accuracy (cross-validation).

3.3 Breast Cancer Wisconsin Diagnostic Data Set

For the BCWDDS data, the results using AB+wLDA are comparable to AB+wLDA_Liu in both training and testing sets, as shown in Figure 3. Moreover, AB+ST performed close to the other two methods at tests, as seen in the cross validation at the bottom of Figure 3, but underperformed on the training set.

The BCWDDS data have a feature vector of size 32. For AB+wLDA and AB+wLDA_Liu, the best possible combination of three features is selected at each iteration. While in the iris data AB+wLDA and AB+ST had the lowest training error, on BCWDDS data AB+wLDA and AB+wLDA_Liu performed with much lower training error. The difference on performance this data is likely related to the difference in features space.

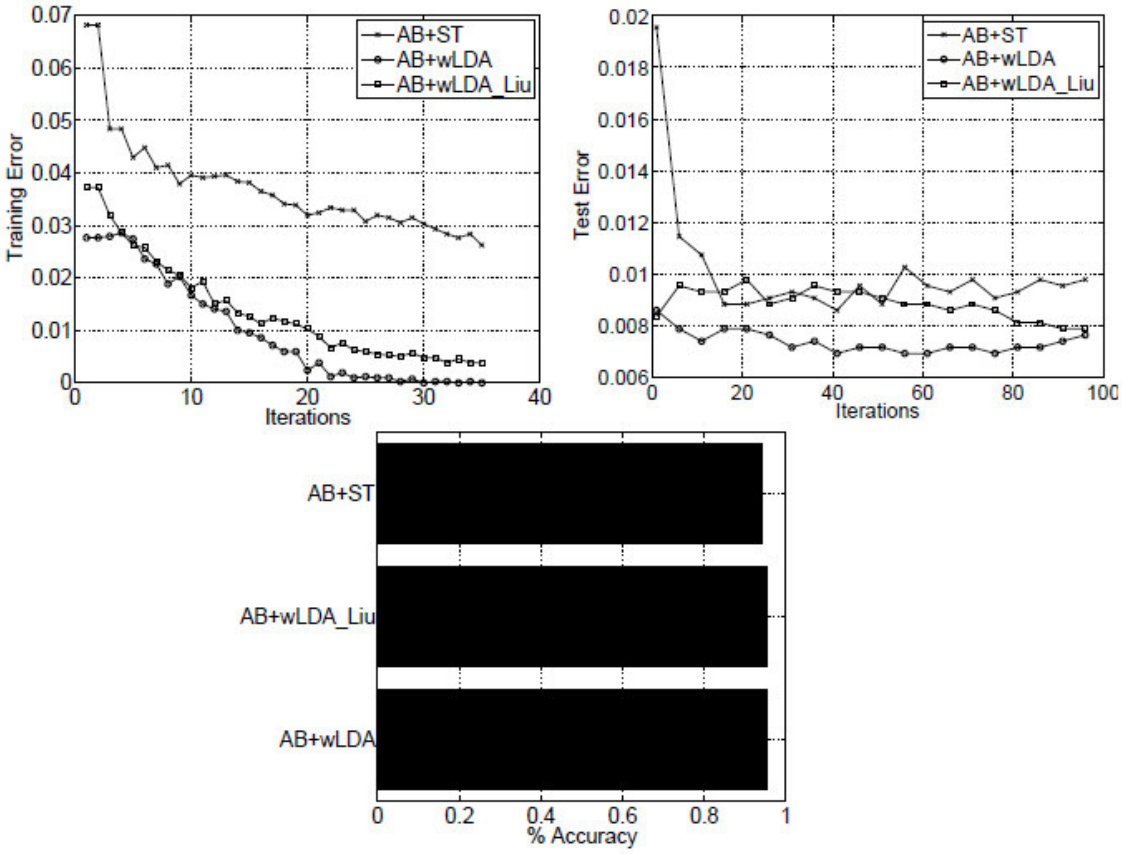

Figure 3. Experimental results on the BCWDDS data. The top row presents training and test errors for each trial run of LKOCV, while the bottom row shows the average accuracy

4. Discussion and Conclusion

This paper proposes a novel method to boosting weighted LDA as weak classifiers that combines the strength and robustness of AdaBoost with LDA. The experimental results demonstrated the advantage of our method over the original Adaboost and a

similarly weighted LDA-based Adaboost with unbalanced scatter matrices, proposed previously by Liu [9]. Our method outperformed both baseline methods using the iris data set while performing comparably with other classifiers when tested on the BCWDDS data set.

The weighted LDA algorithm presented here was proven to be equivalent to the traditional LDA in the case of uniform weight distributions. This observation is important because AdaBoost initializes sample weights uniformly and it is critical to have a true LDA classifier as the initial starting point. By fully incorporating weights into all aspects of the LDA formulation, AB+wLDA was effective in producing an ensemble of rules that can achieve high accuracy, even in low feature space data, such as seen in Figure 1.

Combining Adaboost with LDA allows selecting the most relevant features for classification at each boosting iteration, thus benefiting from feature correlation. The advantages of this approach include the use of a smaller number of weak learners to achieve a low error rate, improved classification performance due to the robustness and stable nature of LDA, and computational efficiency.

AB+wLDA_Liu failed to produce a good ensemble of rules on the artificial data, while AB+ST and AB+wLDA performed well and achieved good separation. This data set had a small feature space, so AB+ST achieved lower training/test error faster than AB+wLDA. AB+wLDA. The results on the Iris Data Set (non-medical) and the Breast Cancer Wisconsin Diagnostic Data Set (medical), both with higher feature space, showed that the combination of Adaboost and weighted linear discriminant analysis outperforms the other classifiers in all the performance metrics used. This confirms the hypothesis that AB+wLDA can achieve low classification error with fewer number of rules, resulting in a more compact and efficient classification model.

The classification method combining Adaboost and weighted LDA is likely to be reproducible on other medical (and non-medical) data sets, making it a valuable tool for clinical diagnosis. For future work, the classification method will be extended for multi-class problems and evaluated on additional data sets in comparison with other classifiers, such as support vector machines.

References

1. Bi J, Periaswamy S, Okada K, Kubota T, Fung G, Salganioff M, Rao RB. Computer aided detection via asymmetric cascade of sparse hyperplane classifiers, Proceedings SIGKDD 2006; 837- 844.
2. Duda R O, Hart P E., Stork D G. Pattern Classification. Wiley-Interscience Publication 2000.
3. Fisher R A. The use of multiple measurements in taxonomic problems. Annual Eugenics 1936; 7(II):179-188.
4. Flores A, Rysavy S, Enciso R, Okada K. Noninvasive differential diagnosis of dental periapical lesions in cone-beam CT. In Proceedings IEEE ISBI 2009; 566-569.
5. Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. In European Conference on Computational Learning Theory 1995; 23–37.
6. Huang Y L, Wang K L. Diagnosis of breast tumors with ultrasonic texture analysis using support vector machines. Neural Computing and Applications 2006; 15(2):164–169.
7. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings IJCAI 1995; 1137–1145.
8. Li X, Wang L, Sung E. Adaboost with SVM-based component classifiers, Engineering Applications of Artificial Intelligence 2008: 21(5): 785-795
9. Liu X, Zhang L, Li M, Zhang H, Wang D. Boosting image classification with LDA-based feature combination for digital photograph management. Pattern Recognition 2005; 38(6): 887–901.

10. Morra JH, Tu Z, Apostolova JG, Green AE, Toga AW, Thompson PM. Comparison of AdaBoost and Support vector machines for deteting Alzheimer'd disease through automated hippocampal segmentation, IEEE Transactions on Medical Imaging 2009; 29(1):30-43.
11. Skurichina M, Duin RPW. Boosting in Linear Discriminant Analyses, Proceedings Int Workshop on Multiple Classifier Systems 2000: 190-199.
12. McCall J O, Wald S S. Clinical Dental Radiology. Philadelphia: Saunders 1954; 4th ed.
13. Tang E K, Suganthan P N, Yao X, Qin A K. Linear dimensionality reduction using relevance weighted LDA. Pattern Recognition 2005; 38: 485-493.
14. Truyen TT, Phung DQ, Venkatesh S, Bui HH. AdaBoost.MRF: Boosted Markov Random Forests and Application to Multilevel Activity Recognition, Proceedings CVPR 2006:1686-1693
15. Viola P, Jones MJ, Snow D. Detecting Pedestrians Using Patterns of Motion and Appearance, International Journal of Computer Vision 2005; 63(2):153-161.
16. Wolberg WH, Street WN, Mangasarian OL. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 1994; 77:163-171.
17. http://archive.ics.uci.edu/ml/index.html.

