Boosting in Linear Discriminant Analysis

Marina Skurichina and Robert P.W.Duin

Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences,
Delft University of Technology, P.O. Box 5046, 2600GA Delft, The Netherlands
Phone: +(31) 15 2783538, FAX: +(31) 15 2786740
{marina, duin}@ph.tn.tudelft.nl

Abstract. In recent years, together with bagging [5] and the random subspace method [15], boosting [6] became one of the most popular combining techniques that allows us to improve a weak classifier. Usually, boosting is applied to Decision Trees (DT's). In this paper, we study boosting in Linear Discriminant Analysis (LDA). Simulation studies, carried out for one artificial data set and two real data sets, show that boosting might be useful in LDA for large training sample sizes while bagging is useful for critical training sample sizes [11]. In this paper, in contrast to a common opinion, we demonstrate that the usefulness of boosting does not depend on the instability of a classifier.

1 Introduction

When data are highly dimensional, having small training sample sizes compared to the data dimensionality, it may be difficult to construct a good single classification rule. Usually, a classifier, constructed on small training sets is biased and has a large variance. Consequently, such a classifier may have a poor performance [1]. In order to improve a weak classifier by stabilizing its decision, a number of techniques could be used, for instance, regularization [2] or noise injection [3].

Another approach is to construct many weak classifiers instead of a single one and combine them in some way into a powerful decision rule. Recently a number of such combining techniques have been developed. The most popular ones are bagging [5], boosting [6] and the random subspace method [15]. In bagging, one samples the training set, generating random independent bootstrap replicates [4], constructs the classifier on each of these and aggregates them by a simple majority vote in the final decision rule. In boosting, classifiers are constructed on weighted versions of the training set, which are dependent on previous classification results. Initially, all objects have equal weights, and the first classifier is constructed on this data set. Then, weights are changed according to the performance of the classifier. Erroneously classified objects get larger weights and the next classifier is boosted on the reweighted training set. In this way a sequence of training sets and classifiers is obtained, which are then combined by a simple majority vote or by a weighted majority vote in the final decision. In the random subspace method classifiers are constructed in random subspaces of the data feature space. Then, only classifiers with the zero classification error on the training set are combined by simple majority vote in the final decision rule.

Usually, bagging, boosting and the random subspace method are applied to DT's [7],[8],[9],[10],[15], where they often produce an ensemble of classifiers, which is superior to a single classification rule. However, these techniques may also perform

well for other classification rules, than DT's. For instance, it was shown that bagging and boosting may be useful for perceptrons (see, e.g. [16]). It was demonstrated that bagging may be beneficial in LDA for small and critical training sample sizes (when the number of training objects is comparable with data dimensionality) [11]. Our initial study [17] has shown that also boosting may be advantageous in LDA.

In this paper we intend to study the usefulness of boosting for linear classifiers and in particular to investigate its relation with the instability of classifiers. We consider the nearest mean classifier [12], the Fisher Linear Discriminant function (FLD) [12] and the regularized FLD [2]. This choice is made in order to observe many different classifiers with a dissimilar instability and, by that, to establish whether the usefulness of boosting depends on the classifier instability or on other classifier peculiarities. The chosen classification rules and their instability are discussed in section 4. One artificial data set and two real data sets representing the 2-class problem are used in our simulation study. They are described in section 3, but first a short description of the boosting algorithm is given in section 2. Simulation results on the performance of boosting in LDA are discussed in section 5. Conclusions are summarized in section 6.

2 The Boosting Algorithm

Boosting, proposed by Freund and Schapire [6], is a technique to combine weak classifiers, having a poor performance, in a strong classification rule with a better performance. As it was already mentioned before, in boosting, classifiers and training sets are obtained sequentially, in a strictly deterministic way. At each step, training data are reweighted in such way that incorrectly classified objects get larger weights in a new modified training set. By that, one actually maximizes margins between training objects. It suggests the connection between boosting and Vapnik's Support Vector Classifier (SVC) [7],[13], as objects obtaining large weights may be the same as the support objects. Boosting is organized by us in the following way.

1. Repeat for b=1,2,...,B.
   a) Construct the classifier $C^b(\boldsymbol{X}^*)$ on the weighted version $\boldsymbol{X}^* = (w_1^b \boldsymbol{X}_1, w_2^b \boldsymbol{X}_2, \ldots, w_n^b \boldsymbol{X}_n)$ of training data set $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$, using weights w_i^b, i=1,...,n ($w_i^b = 1$ for b=1).
   b) Compute probability estimates of the error $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$, $\xi_i^b = \begin{cases} 0, & if\ \ \boldsymbol{X}_i\ \ is\ \ classified\ \ correctly \\ 1, & otherwise \end{cases}$, and $c_b = \frac{1}{2}\log\left(\frac{1-err_b}{err_b}\right)$.
   c) If $0 < err_b < 0.5$, set $w_i^{b+1} = w_i^b \exp(-c_b \xi_i^b)$, i=1,...,n, and renormalize so that $\sum_{i=1}^{n} w_i^{b+1} = n$. Otherwise, set all weights $w_i^b = 1$, i=1,...,n, and restart.
2. Combine classifiers $C^b(\boldsymbol{X}^*)$ by the weighted majority vote with weights c_b to a final decision rule.

3 Data

One artificial data set and two real data sets are used for our experimental study. The first set is a 30-dimensional *correlated Gaussian data* set (*Data I*) constituted by

two classes with equal covariance matrices. Each class consists of 500 vectors. The mean of the first class is zero for all features. The mean of the second class is equal to 3 for the first two features and equal to 0 for all other features. The common covariance matrix is a diagonal matrix with a variance of 40 for the second feature and a unit variance for all other features. The intrinsic class overlap (Bayes error) is 0.064. This data set is rotated using a 30×30 rotation matrix which is $\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ for the first two features and the identity matrix for all other features.

Two real data sets are taken from the UCI Repository [14]. The first is the 34-dimensional *ionosphere* data set (*Data II*) with 225 and 126 objects belonging to the first and the second data class, respectively. The second is the 8-dimensional *diabetes* data set (*Data III*) consisting of 500 and 268 objects from the first and the second data class, respectively. These two data sets were also used in [8], when studying bagging and boosting for decision trees. The diabetes data set was also used when bagging and boosting were studied for LDA [8].

Training sets with 3 to 400, with 3 to 100 and with 3 to 200 objects per class are chosen randomly from a total set for the data I, II and III, respectively. The remaining data are used for testing. All experiments are repeated 50 times for independent training sets. In all figures the averaged results over 50 repetitions are presented. The standard deviations of the mean generalization errors for single and boosted linear classifiers are of the similar order for each data set. When increasing the training sample size, they are decreasing approximately from 0.015 to 0.004, from 0.014 to 0.007 and from 0.018 to 0.004 for the data I, II and III, respectively. When the mean generalization error of the boosted regularized FLD shows a peaking behaviour on the ionosphere data set (see Fig. 4), its standard deviation is about 0.03.

4 The Performance and the Instability of Linear Classifiers

In order to study a large group of linear classifiers and their instability, let us consider regularized classifiers in LDA.

The *Regularized Fisher Linear Discriminant* function (RFLD) [2] is defined as

$$g_{RFLD}(\boldsymbol{x}) = \left[\boldsymbol{x} - \frac{1}{2}(\overline{\boldsymbol{X}}^{(1)} + \overline{\boldsymbol{X}}^{(2)})\right]' (\boldsymbol{S} + \lambda \boldsymbol{I})^{-1} (\overline{\boldsymbol{X}}^{(1)} - \overline{\boldsymbol{X}}^{(2)}) \ ,$$

where the ridge estimate $\boldsymbol{S} + \lambda \boldsymbol{I}$ is used instead of the mean class covariance matrix $\boldsymbol{S}$. One can see, that the RFLD represents a large family of linear classifiers (see Fig. 1). When $\lambda = 0$, one obtains the *Fisher Linear Discriminant* function (FLD) [12]

$$g_{FLD}(\boldsymbol{x}) = \left[\boldsymbol{x} - \frac{1}{2}(\overline{\boldsymbol{X}}^{(1)} + \overline{\boldsymbol{X}}^{(2)})\right]' \boldsymbol{S}^{-1} (\overline{\boldsymbol{X}}^{(1)} - \overline{\boldsymbol{X}}^{(2)}).$$

When $\lambda \to \infty$, the information concerning covariances between features is lost. Then, the classifier approaches the *Nearest Mean Classifier* (NMC) [12]

$$g_{NMC}(\boldsymbol{x}) = \left[\boldsymbol{x} - \frac{1}{2}(\overline{\boldsymbol{X}}^{(1)} + \overline{\boldsymbol{X}}^{(2)})\right]' (\overline{\boldsymbol{X}}^{(1)} - \overline{\boldsymbol{X}}^{(2)}) \ ,$$

and the probability of misclassification may appreciably increase. Small values of the regularization parameter λ may stabilize the decision and improve the classifier performance. However, for very small λ, the effect of regularization will be neglible. In this case the RFLD performs similar to the *Pseudo Fisher Linear Discriminant* (PFLD) [12], having a high classification error around the critical training sample sizes, when the number of training objects is comparable to the data dimensionality.

In order to understand better, when boosting can be beneficial, it is useful to

consider the instability of a classifier [11]. The classifier instability is measured by us by calculating the changes in classification of a training set caused by the bootstrap replicate of the original learning data set. Repeating this procedure several times on the training set (we did it 25 times) and averaging the results, an estimate of the classifier instability is obtained. The mean instability of linear classifiers (on 50 independent training sets) defined in this way is presented in Fig. 2. One can see that the instability of the classifier decreases when the training sample size increases. The instability and the performance of a classifier are correlated: more stable classifiers perform better than less stable ones. In this example, however, the performance of the NMC does not

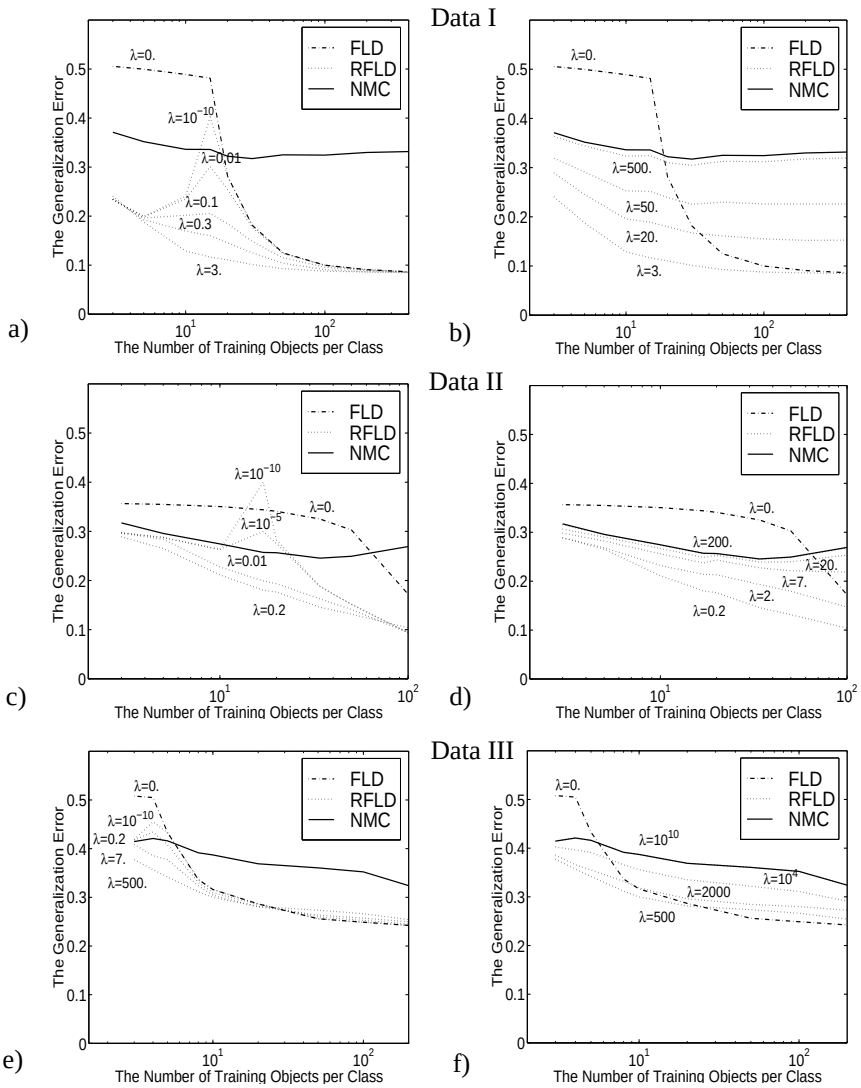

Fig. 1. The performance of the RFLD with different values of λ for Gaussian correlated data (Data I) (a,b), for ionosphere data set (Data II) (c,d) and for diabetes data set (Data III) (e,f)

depend on the training sample size. In contrast to other classifiers, it remains a weak classifier for large training sample sizes, while its stability increases. Theory of boosting is developed for weak classifiers and large training sample sizes. Therefore, one may expect that boosting may be beneficial for the NMC.

5 Boosting for Linear Classifiers

Let us now consider the performance of boosting in LDA on the example of the NMC, the FLD and the RFLD with different values of regularization parameter λ.

The NMC. Boosting is useful for the NMC (see Fig. 3f, Fig. 4f and Fig. 5f). Especially it performs nicely for the Gaussian correlated data set, reducing the general-

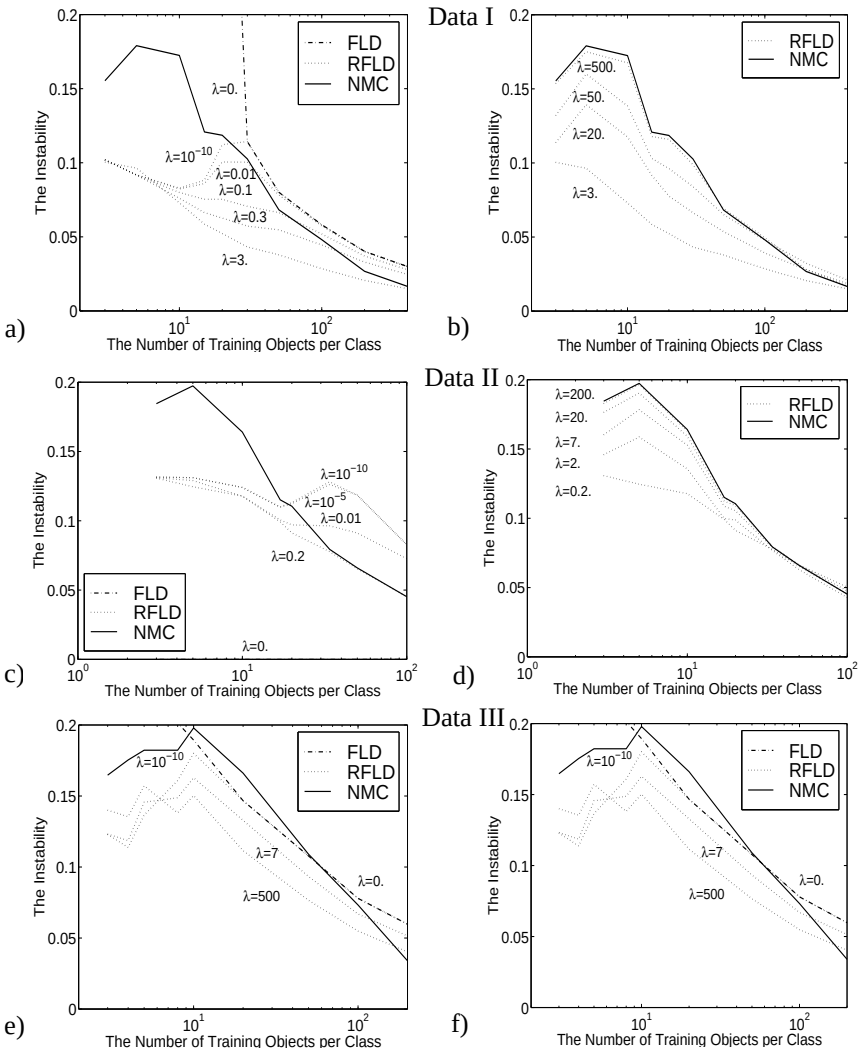

Fig. 2. The instability of the RFLD with different values of λ for Gaussian correlated data (Data I) (a,b), for ionosphere data set (Data II) (c,d) and for diabetes data set (Data III) (e,f)

ization error of a single NMC more than twice. In boosting, wrongly classified objects get larger weights. Mainly, they are objects on the border between classes. Therefore, boosting performs best for large training sample sizes, when the border between data classes becomes more informative. In this case, boosting the NMC performs similar to the linear SVC [13]. However, when the training sample size is large, the NMC is stable. It puts us on the observation that, in contrast to bagging, the usefulness of boosting may not depend directly on the stability of the classifier. It depends on the "quality" of the erroreously classified objects (usually, around the border between data classes) and on the ability of the classifier (its complexity) to distinguish them correctly.

The FLD. Simulation results (see Fig. 3a, Fig. 4a, Fig. 5a) show that boosting is

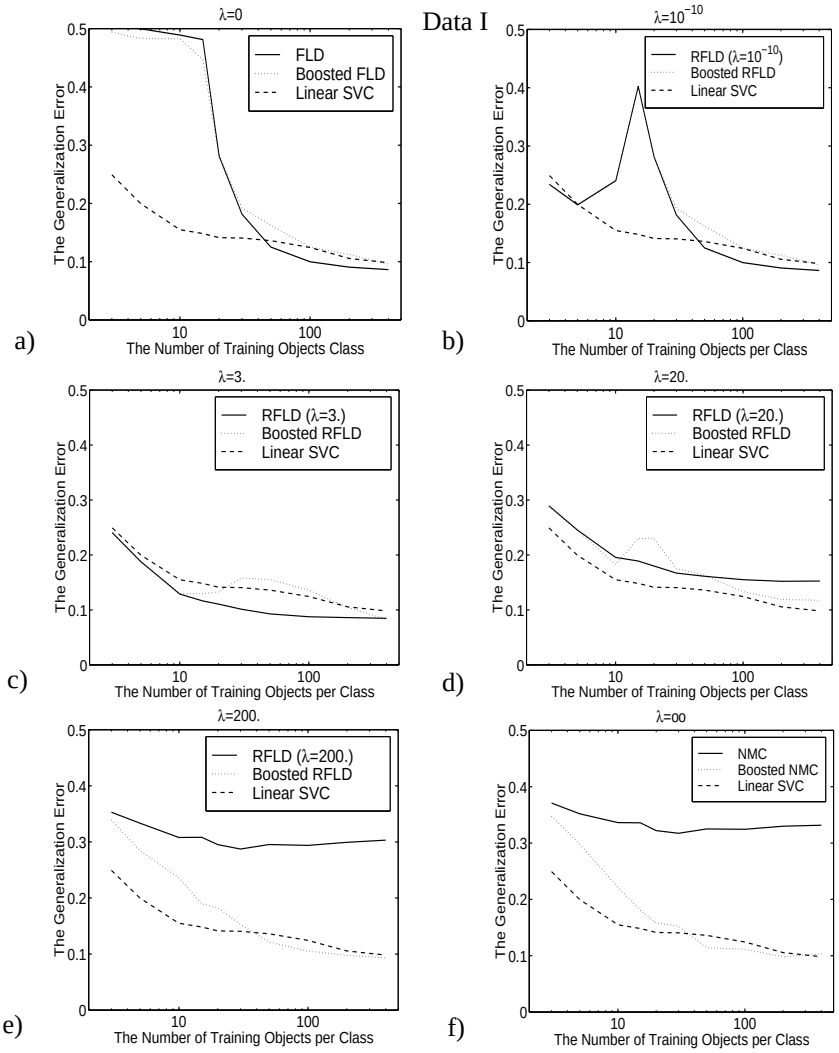

Fig. 3. The performance of the boosting (B=250) for linear classifiers on *Gaussian correlated* data (Data I). Boosting becomes useful, when increasing regularization and the RFLD becomes similar to the NMC

completely useless for the FLD. The performance and the stability of the FLD depends on the training sample size. For small training sample sizes, the classifier is very unstable and has a poor performance, as sample estimates of means have a large bias and a sample estimate of a common covariance matrix is singular or nearly singular. When increasing the training sample size, the sample estimates are less biased, and the classifier becomes more stable and performs better. In boosting, objects on the border between data classes get larger weights. By that, the number of actually used training objects decreases. When the training sample size is smaller than the data dimensionality, all or almost all objects lie on the border. Therefore, almost all training objects are

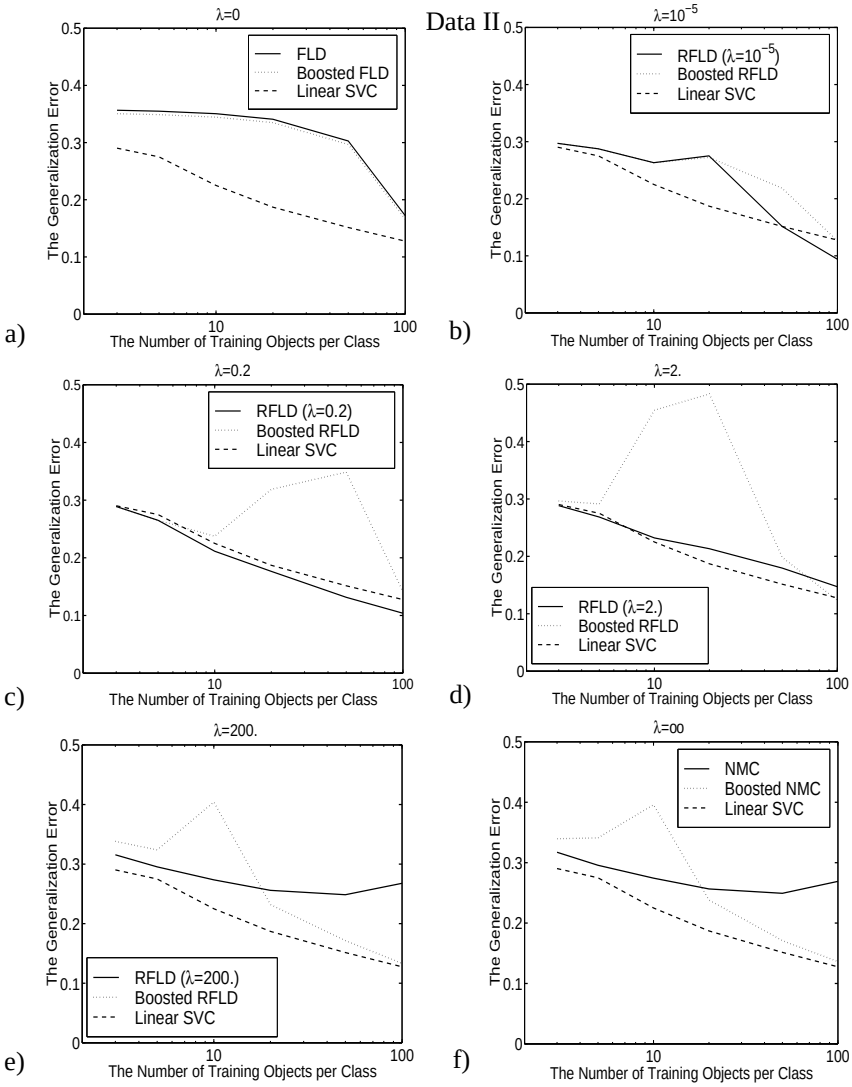

Fig. 4. The performance of boosting (B=250) for linear classifiers on *ionosphere* data (Data II). Boosting becomes useful, when increasing regularization and the RFLD becomes similar to the NMC

used at each step of the boosting algorithm. One gets many similar classifiers that perform badly. Combining such classifiers does not improve the FLD. When the training sample size increases, the FLD performs better. In this case, boosting may perform similar to a single FLD (if the number of objects on the border is sufficiently large to construct a good FLD) or may worsen the situation (if the number of actually used training objects at each step of boosting is not sufficiently large to define a good FLD).

The PFLD. Boosting the PFLD, which is similar to the RFLD with a very small value of the regularization parameter λ, is also useless (see Fig. 3b, Fig. 4b, Fig. 5b). For the training sample sizes larger than the data dimensionality the PFLD, maximiz-

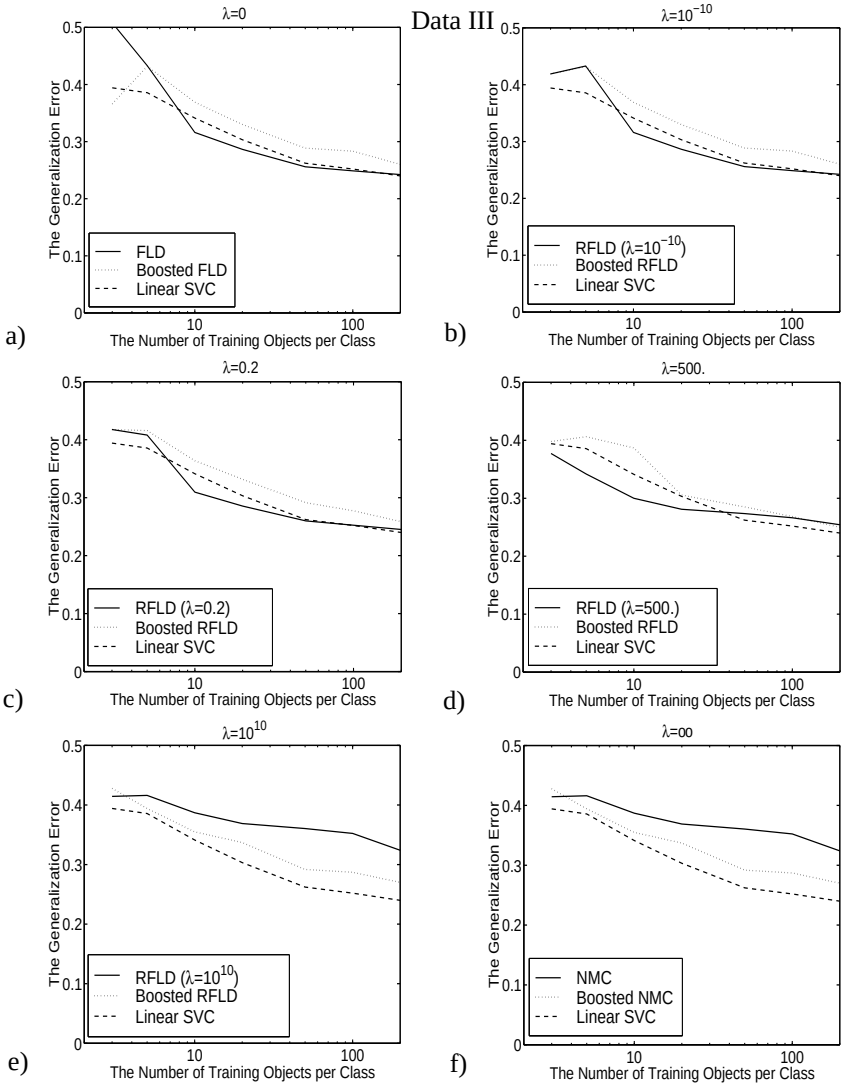

Fig. 5. The performance of boosting (B=250) for linear classifiers for *diabetes* data (Data III). Boosting becomes useful, when increasing regularization and the RFLD becomes similar to the NMC

ing the distance to all given samples, is equivalent to the FLD. For the training sample sizes smaller than the data dimensionality, however, the PFLD finds a linear subspace, which covers all the data samples. On this plane the PFLD estimates the data means and the covariance matrix, and builds a linear discriminant perpendicular to this subspace in all other directions for which no samples are given. Therefore, for these training sample sizes, the apparent error (the classification error on the training set) of the PFLD is always zero. Thus boosting is completely useless for the PFLD.

The RFLD. Considering the RFLD with different values of the regularization parameter λ, one can see that boosting is also not beneficial for these classifiers with exception of the RFLD with very large values of λ, which performs similar to the NMC. For small training sample sizes, when all or almost all training objects have similar weights at each step of the boosting algorithm, the modified training set is similar the original one, and the boosted RFLD performs similar to the original RFLD. For critical training sample sizes, the boosted RFLD may perform worse or even much worse (having a high peak of the generalization error) than the original RFLD. This is caused by two reasons. The first is that the modified training sets used in boosting usually contain less training objects than the original training set. Smaller training sets give more biased sample estimates of classes means and the covariance matrix than larger training sets. Therefore, the RFLD constructed on the smaller training set usually has a worse performance. An ensemble of the worse quality classifiers constructed on the smaller training sets may perform worse than the single classifier constructed on the larger training set. The second reason is that the objects on the border between data classes (which are getting larger weights in the boosting algorithm) have often other distribution than the original training set. Therefore, on such modified training set, the RFLD with certain value of the regularization parameter λ may perform differently than the same RFLD on the original training set. Regularization may not be sufficient, causing the generalization error peak similar to the RFLD with very small values of λ. However, on large training sample sizes, boosting may be beneficial for the RFLD, if the single RFLD performs worse than a linear support vector classifier. As a rule, it is the RFLD with very large values of λ. Thus, boosting is useful only for the RFLD with large values of the regularization parameter λ and for large training sample sizes.

6 Conclusions

Summarizing simulation results presented in the previous section, we can conclude the following:

Boosting may be useful in LDA for classifiers that perform poor on large training sample sizes. Such classifiers are the Nearest Mean Classifier and the Regularized Fisher's Linear Discriminant with large values of the regularization parameter λ, which approximates the NMC.

Boosting is useful only for large training sample sizes, if the objects on the border give a better representation of the distribution of the data classes than the original data classes distribution and the classifier is able (by its complexity) to distinguish them well.

It was shown theoretically and experimentally for DT's [7] that boosting increases the margins of the training objects. By that, boosting is similar to the maximum margin classifiers [13], based on the number of support vectors. In this paper, we have experimentally shown, that boosted linear classifiers may achieve the perfor-

mance of the linear support vector classifier when training sample sizes are large compared with the data dimensionality.

As boosting is useful only for large training sample sizes, when classifiers are usually stable, the performance of boosting does not depend on the instability of the classifier.

The success of boosting depends on many factors including the training sample size, the choice of a weak classifier (the DT, the FLD, the NMC or other), the exact way how the training set is modified, the choice of the combining rule [17] and, finally, the data distribution. By that, it becomes quite difficult to establish universal criteria predicting the usefulness of boosting. Obviously, this question needs more investigation in future.

Acknowledgment

This work is supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO).

References

1. Jain, A.K., Chandrasekaran, B.: Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In: Krishnaiah, P.R., Kanal, L.N. (eds.): Handbook of Statistics, Vol. 2. North-Holland, Amsterdam (1987) 835-855
2. Friedman, J.H.: Regularized Discriminant Analysis. JASA **84** (1989) 165-175
3. An, G.: The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. Neural Computation **8** (1996) 643-674
4. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)
5. Breiman, L.: Bagging predictors. Machine Learning Journal **24**(2) (1996) 123-140
6. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference (1996) 148-156
7. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. The Annals of Statistics **26**(5) (1998) 1651-1686
8. Breiman, L.: Arcing Classifiers. Annals of Statistics, **26**(3) (1998) 801-849
9. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Technical Report (1999)
10. Dietterich, T.G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning, to appear
11. Skurichina, M., Duin, R.P.W.: Bagging for Linear Classifiers. Pattern Recognition **31**(7) (1998) 909-930
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press (1990) 400-407
13. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning **20** (1995) 273-297
14. Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science
15. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(8) (1998) 832-844
16. Avnimelech, R., Intrator, N.: Boosted Mixture of Experts: An Ensemble Learning Scheme. Neural Computation **11** (1999) 483-497
17. Skurichina, M., Duin, R.P.W.: The Role of Combining Rules in Bagging and Boosting. Submitted to S+SSPR 2000, Alicante, Spain