Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Size on Diversity and Accuracy

Marina Skurichina[1], Liudmila I. Kuncheva[2] and Robert P.W. Duin[1]

[1] Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences, Delft University of Technology, P.O. Box 5046, 2600GA Delft, The Netherlands
{marina,duin}@ph.tn.tudelft.nl
[2] School of Informatics, University of Wales, Bangor, Gwynedd, LL57 1UT, United Kingdom
l.i.kuncheva@bangor.ac.uk

Abstract. In combining classifiers, it is believed that diverse ensembles perform better than non-diverse ones. In order to test this hypothesis, we study the accuracy and diversity of ensembles obtained in bagging and boosting applied to the nearest mean classifier. In our simulation study we consider two diversity measures: the *Q* statistic and the disagreement measure. The experiments, carried out on four data sets have shown that both diversity and the accuracy of the ensembles depend on the training sample size. With exception of very small training sample sizes, both bagging and boosting are more useful when ensembles consist of diverse classifiers. However, in boosting the relationship between diversity and the efficiency of ensembles is much stronger than in bagging.

1 Introduction

Some pattern recognition problems cannot be solved by a single classification rule. This happens when the data distribution is very complex or/and data are high dimensional, having small training sample sizes compared to the data dimensionality [1]. In this case, the combined decision of the ensemble of classifiers can be used in order to improve the performance of a single classification rule [2]. However, it is still quite unclear what classifiers the ensemble should consist of in order to be the most effective. An intuitively desirable characteristic of a classifier team is diversity (orthogonality, complementarity, independence etc.) [2, 3]. Theoretically, combining independent classifiers by majority voting will outperform the single classifier. Combining dependent classifiers may be either better or worse than the single classification rule [4]. Therefore, diverse ensembles seem to have a better potential for improving the accuracy than non-diverse ensembles.

In this paper we intend to test this hypothesis on bagging [5] and boosting [6] applied to the Nearest Mean Classifier (NMC) [7]. We choose these combining techniques for our study because they show a good performance on various data sets [8, 9, 10] and provide useful algorithms for constructing ensembles of classifiers. The NMC is chosen for its simplicity, and because both bagging and boosting have been found to be successful for this linear classifier [11]. The performance and the stability of the combined decision in bagging and boosting, applied to linear classifiers, strongly depend on the training sample size [11]. Therefore, in this paper we will study the relationship between accuracy and diversity of ensembles in bagging and boosting with respect to the training sample size.

F. Roli and J. Kittler (Eds.): MCS 2002, LNCS 2364, pp. 62-71, 2002.

The paper is organized in the following way. Section 2 shortly describes bagging and boosting. The two diversity measures used are introduced in section 3. The data sets and the experimental setup are described in section 4. The results of our simulation study are discussed in section 5. Conclusions are summarized in section 6.

2 Combining Techniques

Bagging and boosting are ensemble design techniques that allow us to improve the performance of weak classifiers. Originally, they were designed for decision trees [5, 6]. However, they were found to perform well for other classification rules: neural networks [12], linear classifiers [11] and k-nearest neighbour classifiers [5]. It was shown that for linear classifiers, the performance of bagging and boosting is affected by the training sample size, the choice of the base classifier and the choice of the combining rule [11]. Bagging is useful for linear classifiers constructed on critical training sample sizes, i.e., when the number of training objects is about the data dimensionality. On the other hand, boosting is effective for low-complexity classifiers constructed on large training sample sizes [11, 13]. Both bagging and boosting modify the training data set, build classifiers on these modified training sets and then combine them into a final decision. Usually the simple majority voting is used to get a final decision. However, the weighted majority voting used in boosting [6] is preferable because it is more resistant to overtraining than other combining rules when increasing the number B of combined classifiers [14]. Therefore, we use the weighted majority vote in both studied combining techniques.

Bagging is proposed by Breiman [5] and based on bootstrapping [15] and aggregating concepts thereby benefiting from both approaches. Bootstrapping is based on random sampling with replacement. We take B bootstrap replicates $\boldsymbol{X}^b = (\boldsymbol{X}_1^b, \boldsymbol{X}_2^b, \ldots, \boldsymbol{X}_n^b)$ of the training set $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$ and build a classifier on each of them. *Aggregating* actually means combining the classifiers. Often a combined classifier gives better results than individual classifiers. In bagging, bootstrapping and aggregating techniques are implemented in the following way.

1. Repeat for $b = 1, 2, \ldots, B$.
   a) Take a bootstrap replicate $\boldsymbol{X}^b$ of the training data set $\boldsymbol{X}$.
   b) Construct a classifier $C^b(\boldsymbol{x})$ on $\boldsymbol{X}^b$ with a decision boundary $C^b(\boldsymbol{x}) = 0$[1].
   c) Compute combining weights $c_b = \frac{1}{2}\log\left(\frac{1-err_b}{err_b}\right)$, where $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$ and $\xi_i^b = \begin{cases} 0, \textit{ if } \boldsymbol{X}_i \textit{ is classified correctly} \\ 1, \textit{otherwise} \end{cases}$.
2. Combine classifiers $C^b(\boldsymbol{x})$, $b = 1, \ldots, B$, by the weighted majority vote with weights c_b to a final decision rule $\beta(x) = \begin{cases} +1, & \textit{if} \quad \sum_b c_b \operatorname{sgn}(C^b(\boldsymbol{x})) > 0 \\ -1, & \textit{otherwise} \end{cases}$.

Boosting, proposed by Freund and Schapire [6], is another technique to combine weak classifiers having a poor performance in order to get a classification rule with a better performance. In boosting, classifiers and training sets are obtained in a strictly *deterministic* way. Both, training data sets and classifiers are obtained

[1] We note that both bagging and boosting were originally defined for two classes. The class labels for the objects are therefore encoded as -1 and +1, with $C^b(\boldsymbol{x}) = 0$ being the decision boundary.

sequentially in contrast to bagging, where training sets and classifiers are obtained *randomly* and *independently* (in parallel) from the previous step of the algorithm. At each step of boosting, training data are reweighted in such a way that incorrectly classified objects get larger weights in a new, modified training set. By that, one actually maximizes the margins of the training objects (the distance of the training object to the decision boundary). By this, boosting is similar Vapnik's Support Vector Classifier (SVC) [16]. However, in boosting, margins are maximized locally for each training object, while in the support vector classifier, global optimization is performed.

In this study, boosting is organized in the following way. It is based on the "arc-fs" algorithm described by Breiman [17], where we reweight the training set instead of resample it. The "arc-fs" algorithm is the improved version of the standard AdaBoost algorithm [13]. Additionally, we set initial weight values w_i^1, $i = 1, \ldots, n$, to 1 instead of $1/n$, in order to be independent of data normalization. Therefore, boosting is implemented by us as follows.

1. Repeat for $b = 1, \ldots, B$.

   a) Construct a base classifier $C_b(\boldsymbol{x})$ (with a decision boundary $C_b(\boldsymbol{x})=0$) 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, $i = 1, \ldots, n$ (all $w_i^b = 1$ for $b = 1$).

   b) Compute combining weights $c_b = \frac{1}{2}\log\left(\frac{1-err_b}{err_b}\right)$, where $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$ and $\xi_i^b = \begin{cases} 0, if\ \boldsymbol{X}_i\ is\ classified\ correctly \\ 1, otherwise \end{cases}$.

   c) If $0 < err_b < 0.5$, set $w_i^{b+1} = w_i^b \exp(c_b \xi_i^b)$, $i = 1, \ldots, n$, and renormalize so that $\sum_{i=1}^{n} w_i^{b+1} = n$. Otherwise, restart the algorithm with weights $w_i^b = 1$, $i = 1, \ldots, n$.

2. Combine base classifiers $C_b(\boldsymbol{x})$, $b = 1, \ldots, B$, by the weighted majority vote with weights c_b to a final decision rule $\beta(x) = \begin{cases} +1, & if \quad \sum_b c_b \mathrm{sgn}(C^b(\boldsymbol{x})) > 0 \\ -1, & otherwise \end{cases}$.

3 Diversity Measures

Different diversity measures are introduced in the literature [18]. In our study, we consider the *Q* statistic and the disagreement measure.

The Q statistic is the pairwise symmetrical measure of diversity proposed by Yule [19]. For two classifiers C_i and C_j, *Q* statistic is defined as

$$Q_{ij} = \frac{ad - bc}{ad + bc},$$

where

a is the probability that both classifiers C_i and C_j make the correct classification,
b is the probability that the classifier C_i is correct and the classifier C_j is wrong,
c is the probability that the classifier C_i is wrong and the classifier C_j is correct,
d is the probability that both classifiers C_i and C_j are wrong,
and $a + b + c + d = 1$.

For a set of *B* classifiers, the averaged statistic *Q* of all pairs (C_i, C_j) is calculated. *Q* varies between -1 and 1. For statistically independent classifiers, it is 0. So, the higher the absolute value of *Q* the less diverse the team of classifiers (denoted $\downarrow$).

The disagreement measure (used in [20, 21]) is defined as

$$D_{ij} = b + c \quad .$$

It is also a pairwise symmetrical measure of diversity. For a set of B classifiers, the averaged statistics D of all pairs is calculated. The higher the value of D the more diverse the team of classifiers (denoted $\uparrow$).

4 Data

In our experimental investigations we considered one artificial and three real data sets representing two-class problems.

The artificial data set, called the *80-dimensional Gaussian correlated data*, was chosen for the many redundant features. The set consists of two classes with equal covariance matrices. Each class is constituted by 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 diagonal with a variance of 40 for the second feature and unit variance for all other features. This data set is rotated in the subspace spanned by the first two features using a rotation matrix $\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$. The intrinsic class overlap, found by Monte Carlo experiments (an estimate of the Bayes error) is 0.07.

The three real data sets are taken from the UCI Repository [22]. They are the 8-dimensional *pima-diabetes* data set, the 34-dimensional *ionosphere* data set and the 60-dimensional *sonar* data set.

Training sets are chosen randomly and the remaining data are used for testing. All experiments are repeated 50 times on independent training sets. So all the figures below show the averaged results over 50 repetitions. The standard deviations of the mean generalization errors for the single and combined decisions are around 0.01 for each data set.

For both bagging and boosting, we choose B=250 classifiers. As explained earlier, the classifiers were combined by the weighted majority voting to reach a final decision.

5 Diversity and Accuracy of Ensembles in Bagging and Boosting

Let us now consider the accuracy and diversity of ensembles in bagging and boosting when the combining techniques are applied to the nearest mean classifier. It has been shown [11] that the performance of bagging and boosting is affected by the training sample size. This is nicely demonstrated by Fig. 1, 2a,b and 3a,b. Bagging is mainly useful for critical training sample sizes when the number of training objects is comparable with the data dimensionality. Boosting performs best for large training sample sizes. Fig. 2c-f and 3c-f show that diversity of classifiers in the ensembles obtained by bagging and boosting also depends on the training sample size.

In bagging, the classifiers are constructed on bootstrap replicates of the training set. Bootstrapping is most effective when the training sample size is smaller or comparable with the data dimensionality. In this case, one obtains bootstrap replicates with the most dissimilar statistical characteristics. Classifiers constructed on such bootstrap replicates will be also diverse. When the training sample size increases, bootstrapping the training set becomes less effective. Bootstrap replicates of large and very large training sets have similar statistical characteristics, because such training sets represent the real data distribution well and perturbations in their composition

barely affect these characteristics. Classifiers constructed on such bootstrap replicates are also similar. Thus, *in bagging, classifiers in the ensemble become less diverse when the number of training objects increases*.

In boosting, the training set is modified in such a way that training objects incorrectly classified at the previous step get larger weights in a new, modified training set. The larger the training set, the higher the number of borderline objects. Consequently, larger training sets are more sensitive than smaller ones to changes in the discriminant function (see example in Fig. 4). In large training sets, more training objects can be classified differently even after a small change in the discriminant function. Consequently, at each step of boosting, more changes in weights for training objects occur. By this, we obtain more diverse training sets and, therefore, more diverse classifiers for large training sample sizes. So, *in boosting, diversity of classifiers in the ensemble increases with an increase in the training sample size*.

Before studying the relationship between accuracy and diversity in bagging and boosting, let us note that the accuracy of the ensemble always depends on the training sample size. Usually the accuracy of statistical classifiers increases with an increase in the number of training objects. Therefore, combining classifiers of a higher accuracy (obtained on larger training sets) may also result in a better combined decision (with a higher accuracy) than when worse performing classifiers (obtained on smaller training sets) are combined. By this, increasing the training sample size, ensembles may perform better whatever the diversity of the ensemble is. As an illustration of this phenomenon let us consider the sonar data set, where bagging is inefficient, i.e., does not improve on the single classifier (see Fig. 3b). We observe, however, that the accuracy of bagging

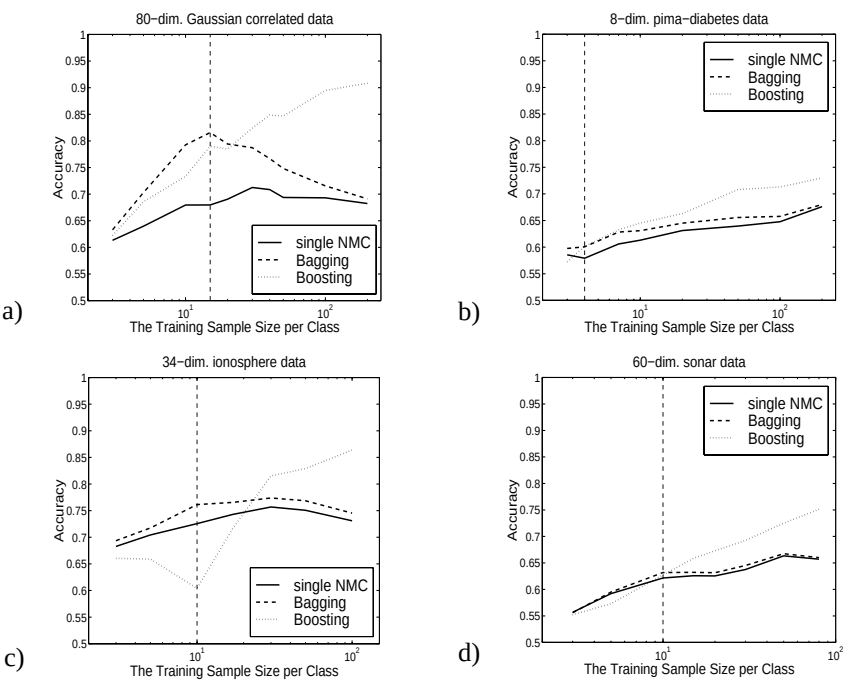
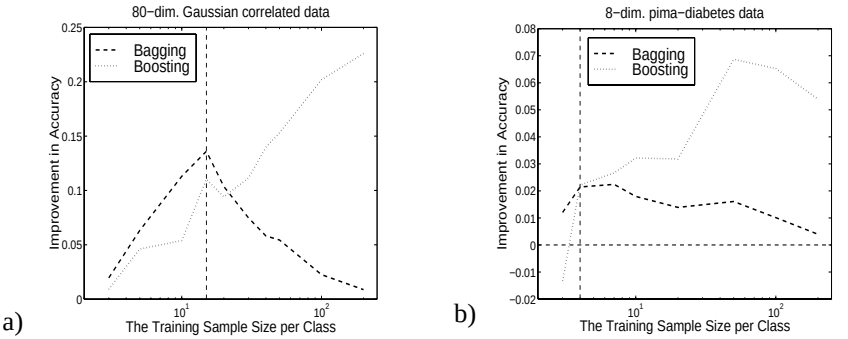


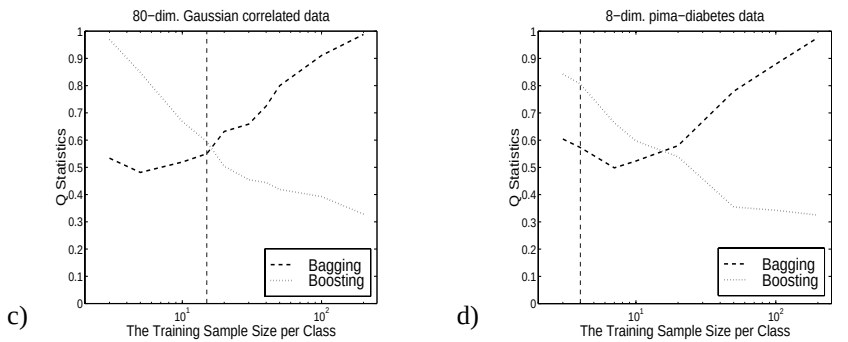
Fig. 1. The accuracy of the single NMC, bagging and boosting applied to the NMC versus the training sample size for the Gaussian correlated, pima-diabetes, ionosphere and sonar datasets.

80-dimensional Gaussian correlated data 8-dimensional pima-diabetes data

The Improvement in Accuracy over the single NMC (↑)



Diversity measured by Q statistics (↓)



Diversity measured by Disagreement measure (↑)

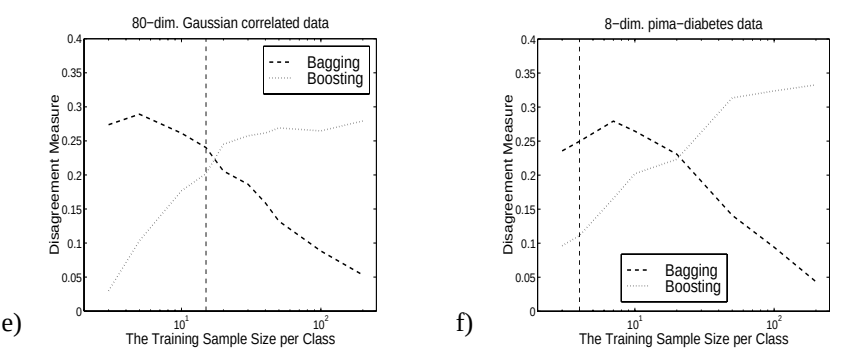


Fig. 2. The efficiency (the improvement in accuracy over the single NMC) and diversity of ensembles (with the NMC as the base classifier) in bagging and boosting versus the training sample size for the 80-dimensional Gaussian correlated data and the 8-dimensional pima-diabetes data (↑ - the larger the better, ↓ - the smaller the better).

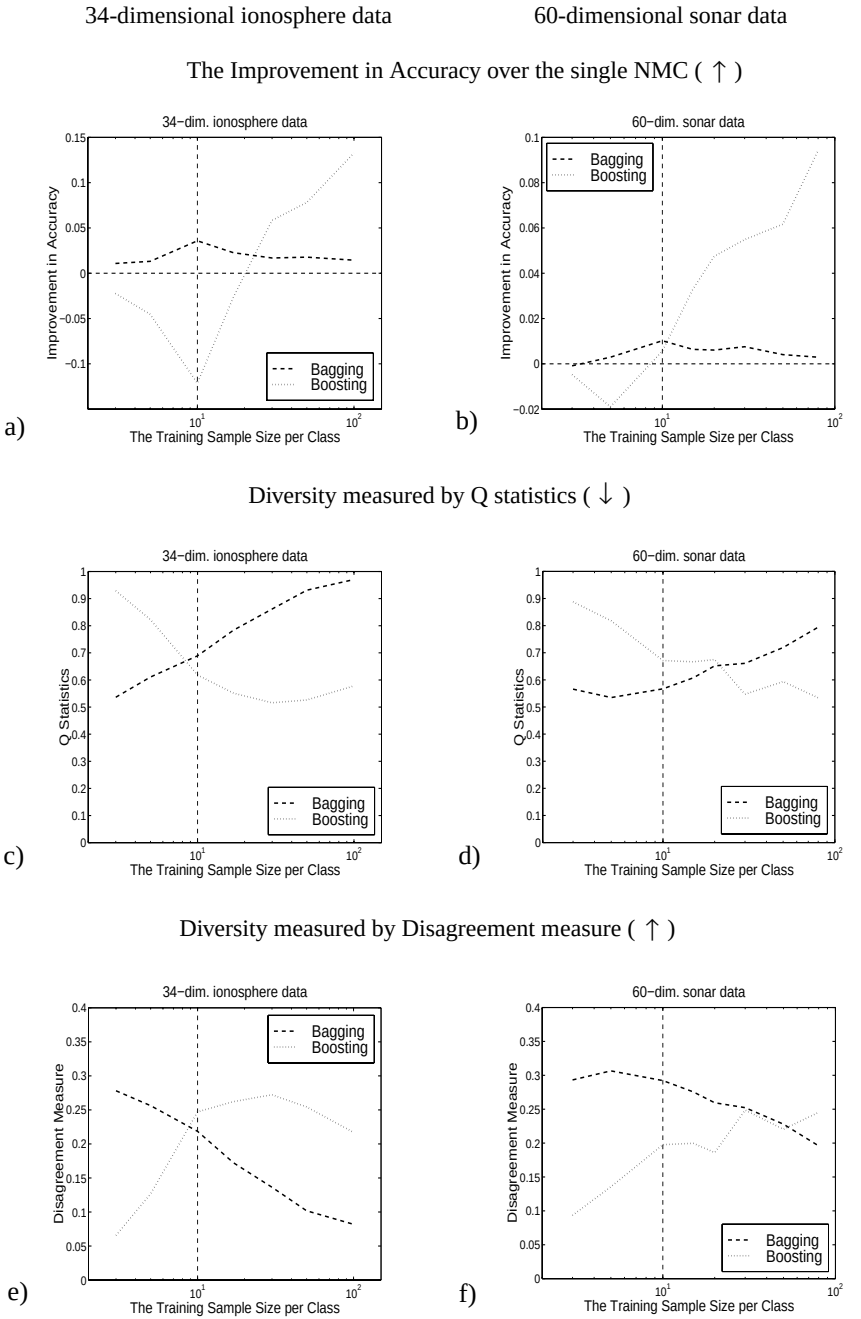


Fig. 3. The efficiency (the improvement in accuracy over the single NMC) and diversity of ensembles (with the NMC as the base classifier) in bagging and boosting versus the training sample size for the 34-dimensional ionosphere data and the 60-dimensional sonar data (↑ - the larger the better, ↓ - the smaller the better).

(see Fig. 1d) increases with an increase in the number of training objects, because the accuracy of the base classifier increases when the number of training objects becomes larger. Boosting also improves when increasing the training sample size. Diversity of the ensembles obtained in bagging and boosting gives us a different insight into the problem (see Fig. 3d,f). We are interested in how the diversity of the ensembles affects the usefulness of the combining techniques. Thus, in our study, instead of studying the accuracy of the combining techniques with respect to diversity of the ensembles, we consider the relationship between the improvement over the performance of the single classifier. The plots in Fig. 2 and 3 help us assess this relationship by eye.

Both diversity and the usefulness of the combining techniques depend on the training sample size. When the training sample size is very small (much smaller than the data dimensionality), the training set is usually non-representative. Modifications by sampling or reweighting of a small and inadequate data set hardly ever give a better representation of the real data distribution. As a result, inaccurate classifiers are obtained all the way through bagging or boosting [1]. Combining these classifiers, be they diverse or non-diverse, is meaningless (see Fig. 1, 2a,b and 3a,b). Therefore, we will exclude very small training sample sizes from our study. Below we discuss only training sample sizes that are not very small: approximately larger than 10 training objects per class for the 80-dimensional Gaussian correlated, the 34 dimensional ionosphere and the 60-dimensional sonar data sets and larger than 4 training objects per class for the 8-dimensional pima-diabetes data set.

Let us now consider diversity of ensembles in bagging and boosting and the efficiency of these combining techniques. Fig. 2 and 3 show that, with exception of very small training sample sizes, *bagging and boosting are more useful (they give a larger improvement over the performance of the single classifier) when they construct diverse ensembles.* When increasing the training sample size, both the efficiency of bagging and diversity of ensembles constructed by bagging decrease. Bagging constructs more diverse ensembles and, therefore, is more useful for the Gaussian correlated and the

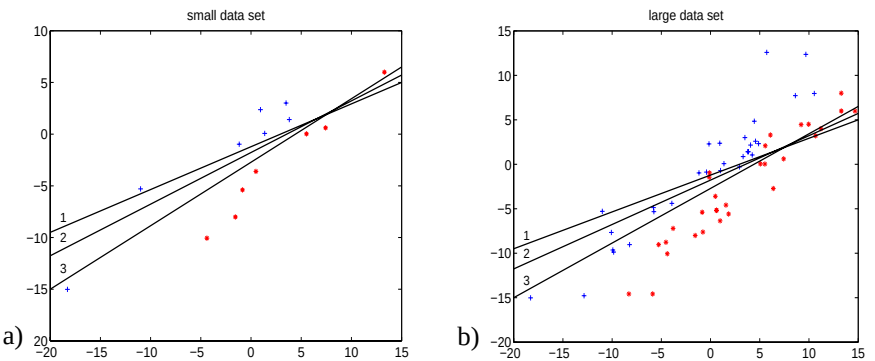


Fig. 4. The example of sensitivity of a small data set (plot a) and a large data set (plot b) to changes in the discriminant function. The classification of the small data set is robust: when the discriminant function changes from 1 to 3, the classification of the data set remains the same. The large training set is more sensitive: when discriminant function changes from 1 to 3, bordeline objects are classified differently. In boosting, when the training set is small (plot a), at each step (1, 2 and 3) the same 2 objects get large weights in a new, modified data set. When the training set is large (plot b), different training objects get large weights at steps 1, 2 and 3. By this, in boosting, we obtain more diverse training sets when the training set is large than when it is small.

pima-diabetes data sets (see Fig. 2) than for the ionosphere data set (see Fig. 3a,c,e). However, the usefulness of bagging does not depend only on diversity of the ensemble. The quality of the constructed base classifier in solving a problem is also important. The sonar data set has a complex data distribution with a non-linear boundary between data classes. Bagging constructs quite diverse classifiers on this data set, but it is not useful (see Fig. 3b,d,f). Boosting constructs more diverse classifiers and becomes more useful when the training sample size increases. Boosting has an advantage to bagging, because by overweighting the borderline objects it constructs classifiers focused on the neighbourhood of a border between data classes. When increasing the training sample size, the border between data classes becomes better defined, which is why boosting constructs more diverse classifiers. Thus, boosting of linear classifiers can be useful even when solving complex problems like in the sonar data set (see Fig, 3b).

As it was mentioned before, the efficiency of the combining techniques depends not only on the diversity of the constructed ensemble but also on the "quality" of the obtained classifiers. Boosting takes special care about the "quality" of classifiers: they are constructed to account for the important regions of the data space, these around the border between classes. Bagging does not take such special care and, therefore, is less efficient for the type of problems considered here. Thus, *in boosting, the relationship between the efficiency and diversity of classifiers is much stronger than in bagging*.

6 Conclusions

In this paper we have studied the relationship between diversity of ensembles in bagging and boosting and the efficiency of these combining techniques applied to the NMC. The accuracy of the ensembles has been obtained by the weighted majority voting. The efficiency of the combining techniques has been measured by the difference between the accuracy of the combined classifier and the accuracy of the single classifier. In our study we have considered two diversity measures: the Q-statistic and the disagreement measure. The simulation study performed on four data sets has shown the following.

Diversity of ensembles in bagging and boosting is affected by the training sample size. In bagging, diversity of ensembles decreases with an increase in the number of training objects. In boosting, the classifiers in the ensemble become more diverse, when the number of training objects increases.

With exception of very small training sample sizes, both bagging and boosting perform better when classifiers in the ensemble are diverse. However, for boosting, the correlation between the efficiency and diversity of ensembles is much stronger than for bagging, because in boosting special care is taken about constructing the classifiers in regions around the borders between data classes.

As the efficiency of the combining techniques is correlated with diversity (especially for boosting), diversity might be useful as a possible criterion for predicting a potential efficiency of the ensemble, for selecting a proper combining rule to aggregate classifiers in the ensemble, or as a stopping criterion against overtraining in bagging and/or boosting.

Acknowledgment

This work is supported by the Dutch Technology Foundation (STW).

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. Lam, L.: Classifier Combinations: Implementations and Theoretical Issues. In: Kittler, J., Roli, F. (eds.): Multiple Classifier Systems (Proc. of the First Int. Workshop MCS, Cagliari, Italy). Lecture Notes in Computer Science, Vol. 1857, Springer-Verlag, Berlin (2000) 78-86
3. Cunningham, P., Carney, J.: Diversity versus Quality in Classification Ensembles Based on Feature Selection. Tech. Report TCD-CS-2000-02, Dept. of Computer Science, Trinity College, Dublin (2000)
4. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W.: Is Independence Good for Combining Classifiers? In: Proc. of the 15th Int. Conference on Pattern Recognition, Vol. 2, Barcelona, Spain (2000) 169-171
5. Breiman, L.: Bagging predictors. In: Machine Learning Journal **24**(2) (1996) 123-140
6. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Machine Learning: Proc. of the 13th Int. Conference (1996) 148-156
7. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press (1990) 400-407
8. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. In: Machine Learning **36** (1999) 105-142
9. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.): Multiple Classifier Systems (Proc. of the First Int. Workshop MCS, Cagliari, Italy). Lecture Notes in Computer Science, Vol. 1857, Springer-Verlag, Berlin (2000) 1-15
10. Quinlan, J.R.: Bagging, Boosting, and C4.5. In: Proc. of the 14th National Conference on Artificial Intelligence (1996)
11. Skurichina, M.: Stabilizing Weak Classifiers. PhD thesis, Delft University of Technology, Delft, The Netherlands (2001)
12. Avnimelech, R., Intrator, N.: Boosting Regression Estimators. In: Neural Computation **11** (1999) 499-520
13. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. In: Journal of Computer and System Sciences **55**(1) (1997) 119-139
14. Skurichina, M., Duin, R.P.W.: The Role of Combining Rules in Bagging and Boosting. In: Ferri, F.J., Inesta, J.M., Amin, A., Pudil, P. (eds.): Advances in Pattern Recognition (Proc. of the Joint Int. Workshops SSPR and SPR, Alicante, Spain). Lecture Notes in Computer Science, Vol. 1876, Springer-Verlag, Berlin (2000) 631-640
15. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman&Hall, New York (1993)
16. Cortes, C., Vapnik, V.: Support Vector Networks. In: Machine Learning **20** (1995) 273-297
17. Breiman, L.: Arcing Classifiers. In: Annals of Statistics **26**(3) (1998) 801-849
18. Kuncheva, L.I., Whitaker, C.J.: Measures of Diversity in Classifier Ensembles (submitted)
19. Yule, G.U.: On the Association of Attributes in Statistics. In: Phil. Transactions **A**(194) (1900) 257-319
20. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. In: IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(8) (1998) 832-844
21. Skalak, D.B.: The Sources of Increased Accuracy for Two Proposed Boosting Algorithms. In: Proc. of American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop (1996)
22. Blake, C.L., Merz, C.J.: 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 (1998)