Using Diversity with Three Variants of Boosting: Aggressive, Conservative and Inverse

Ludmila I. Kuncheva and Christopher J. Whitaker

School of Informatics, University of Wales, Bangor
Bangor, Gwynedd, LL57 1UT, United Kingdom
{l.i.kuncheva,c.j.whitaker}@bangor.ac.uk

Abstract. We look at three variants of the boosting algorithm called here Aggressive Boosting, Conservative Boosting and Inverse Boosting. We associate the diversity measure Q with the accuracy during the progressive development of the ensembles, in the hope of being able to detect the point of "paralysis" of the training, if any. Three data sets are used: the artificial Cone-Torus data and the UCI Pima Indian Diabetes data and the Phoneme data. We run each of the three Boosting variants with two base classifier models: the quadratic classifier and a multi-layer perceptron (MLP) neural network. The three variants show different behavior, favoring in most cases the Conservative Boosting.

1 Introduction

Boosting algorithms are amongst the most popular methods for constructing classifier ensembles [1, 3, 5, 13]. They build the ensemble incrementally, placing increasing weights on those objects in the data set, which appear to be "difficult". The presumption is that this introduces diversity into the ensemble, and therefore enhances the performance. It has been found however, that boosting might get "paralyzed" in the sense that adding more classifiers does not lead to further improvement of the performance [17] and the ensemble testing error might start to increase again.

In this study we are interested in how the diversity of the ensemble progresses when new classifiers are added, one at a time, and how the pattern of diversity is related to the training and testing errors. Section 2 introduces the concept of diversity in classifier ensemble and the Q measure of diversity. In Section 3, three variants of ADAboost (with resampling) are described: Aggressive Boosting, Conservative Boosting and Inverse Boosting. Section 4 gives an illustration of the relationship between the three methods on the one hand, and the measure of diversity Q on the other hand.

2 Diversity in classifier ensembles

Several authors have pointed out the importance of diversity for the success of classifier ensembles [2,6,7,11,12,16]. So far there is no diversity measure accepted by consensus, perhaps owing to the lack of a clear-cut relationship between the measures of diversity and the accuracy of the ensemble [10, 14]. Our previous studies led us to the choice of the Q statistic for measuring diversity [9]. The calculation of Q [18] is based on a pairwise table for each pair of classifiers in the ensemble.

Let $\mathcal{D} = \{D_1, \ldots, D_L\}$ be the ensemble, built on the data set Z, such that $D_i : \Re^n \rightarrow \Omega$ for $\mathbf{x} \in \Re^n$. For each classifier D_i, we record whether it correctly classifies $\mathbf{z}_j$ (the label it produces matches the true label) or not. Consider two classifiers D_i and D_k, and a 2×2 table of probabilities that summarizes their combined outputs as in Table 1.

Table 1. The 2×2 relationship table with probabilities

	D_k correct (1)	D_k wrong (0)
D_i correct (1)	a	b
D_i wrong (0)	c	d

$$Q_{i,k} = \frac{ad - bc}{ad + bc}. \tag{1}$$

Total, $a + b + c + d = 1$

Many pairwise statistics have been proposed as measures of similarity in the numerical taxonomy literature (e.g., [15]). The Q statistic is designed for categorical data with the same intuition as the correlation coefficient for continuous-values data. It is calculated from Table 1 as shown.

For independent D_i and D_k, $Q_{i,k} = 0$. Since independence is important in classifier combination, although not necessarily the best scenario [10], the zero value of Q is a practical target to strive for. We have found that negative values of Q are even better but such ensembles are unlikely to be developed. If we calculate the correlation coefficient between the values 0 (incorrect) and 1 (correct), using the distribution in Table 1, the resultant formula will have the same numerator as Q and a positive (but more cumbersome to calculate) denominator. As with Q, the correlation coefficient will give a value 0 for independence. For none of the other 9 diversity measures researched by us, is there any fixed value for independence [9].

3 The three Boosting variants

The general boosting idea is to develop the classifier team $\mathcal{D}$ incrementally, adding one classifier at a time. The classifier that joins the ensemble at step k is trained on a data set selectively sampled from the training data set Z. The sampling distribution starts from uniform, and progresses towards increasing the likelihood of "difficult" data points. Thus the distribution is updated at each step, increasing the likelihood of the objects misclassified by the classifier at step $k - 1$. The basic algorithm implementing this idea is shown in Figure 1. We use the data set $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ to construct and ensemble of L classifiers.

1. Initialize all coefficients as $W_1(i) = \frac{1}{N}$, $i = 1, \ldots, N$. We start with an empty classifier ensemble $\mathcal{D} = \emptyset$ and initialize the iterate counter $k = 1$.
2. For $k = 1, \ldots, L$

2.1. Take a sample S_k from Z using distribution W_k.
2.2. Build a classifier D_k using S_k as the training set.
2.3. Calculate the weighted ensemble error at step k by

$$\epsilon_k = \sum_{i=1}^{N} W_k(i)(1 - y_{i,k}), \tag{2}$$

where $y_{i,k} = 1$, if D_k correctly recognizes $\mathbf{z}_i \in Z$, and $y_{i,k} = 0$, otherwise. If $\epsilon_k = 0$ or $\epsilon_k \geq 0.5$, the weights $W_k(i)$ are reinitialized to $\frac{1}{N}$.
2.4. Next we calculate the coefficient

$$\beta_k = \sqrt{\frac{1 - \epsilon_k}{\epsilon_k}}, \quad \epsilon_k \in (0, 0.5), \tag{3}$$

to be used in the weighted voting, and subsequently update the individual weights

$$W_{k+1}(i) = \frac{W_k(i)\beta_k^{\xi(y_{i,k})}}{\sum_{j=1}^{N} W_k(j)\beta_k^{\xi(y_{j,k})}} \quad , \quad i = 1, \ldots, N. \tag{4}$$

where $\xi(y_{i,k})$ is a function which specifies which of the three Boosting variants we use.

End k.
3. The final decision for a new object $\mathbf{x}$ is made by weighted voting between the L classifiers. First, all classifiers give labels for $\mathbf{x}$ and then for all D_k that gave label ω_t, we calculate the support for that class by

$$\mu_t(\mathbf{x}) = \sum_{D_k(\mathbf{x}) = \omega_t} \ln(\beta_k). \tag{5}$$

The class with the maximal support is chosen for $\mathbf{x}$.

Fig. 1. A general description of the Boosting algorithm for classifier ensemble design

The three variants of Boosting are as follows:

1. Aggressive Boosting. In this version, the weights for the incorrectly classified objects are increased *and* the weights of the correctly classified objects are decreased at the same step k. Note that even if we do not decrease the weights of the correctly classified objects, they will be decreased anyway by the normalization step. This will happen because we have increased some of the $W_k(i)$'s, and for the sum to be 1, all the remaining weights must go down. The adjective "aggressive" expresses the fact that we force this difference even further. For this case, $\xi(y_{i,k}) = 1 - 2y_{i,k}$. Aggressive Boosting is the versions of ADAboost in [4,13].

2. Conservative Boosting. Here the weights are changed only in one direction: either the weights of the correctly classified objects are decreased, as for example in [1], *or* the weights of the misclassified objects are increased, e.g. [5]. For the latter case, we use $\xi(y_{i,k}) = 1 - y_{i,k}$.

3. Inverse Boosting. This variant is similar to the "hedge" algorithm described in [5]. The philosophy is completely opposite to that of Boosting. Instead of increasing the likelihood of the "difficult" objects, we decrease it, thereby gradually filtering them out. Thus the classifiers will tend to be more and more similar, eliminating any diversity in the process. The idea for this inverse boosting originated by a missprint (we believe) in [5] by which it turns out that the weights of the misclassified objects were actually decreased. We were curious to see whether the opposite strategy lead anywhere, so we brought this variant into the study as well. For the inverse boosting, $\xi(y_{i,k}) = y_{i,k} - 1$.

In a way, variants 1 and 3 are the two extremes, and 2 is a softer version of 1. Note that although all three variants appear in the literature, no particular distinction has been made between them, most of the time all being called AdaBoost.

4 Experiments

The two data sets used are the Cone-torus data[1], and the Pima Indian Diabetes data set from the UCI Machine Learning Repository[2]

Figure 2 (left) displays the averaged results from 10 runs of the three variants on the Cone-Torus data, using quadratic discriminant classifiers as the base classifiers. The training and testing accuracies are plotted versus the number of the classifiers, as they are added one at a time. Underneath each of these plots, the training and testing diversity Q is shown. On the right in Figure 2 are the point likelihoods found through the three Boosting variants. The light gray color corresponds to the higher likelihood.

The following observations can be made

- The three methods give different performance patterns. The Aggressive Boosting shows overtraining after $L = 13$ while the other two methods gradually decrease both training and testing errors in a close correspondence between the two.
- Training and testing diversities are approximately identical for all three methods and have minima indicating a good place to stop the training. In this example, an early stopping is especially important for the Aggressive Boosting because of the overtraining. The Q has a characteristic 'tic'-shape,

[1] available at http://www.bangor.ac.uk/∼mas00a/Z.txt and Zte.txt, for more experimental results see [8]

[2] available at http://www.ics.uci.edu/∼mlearn/MLRepository.html

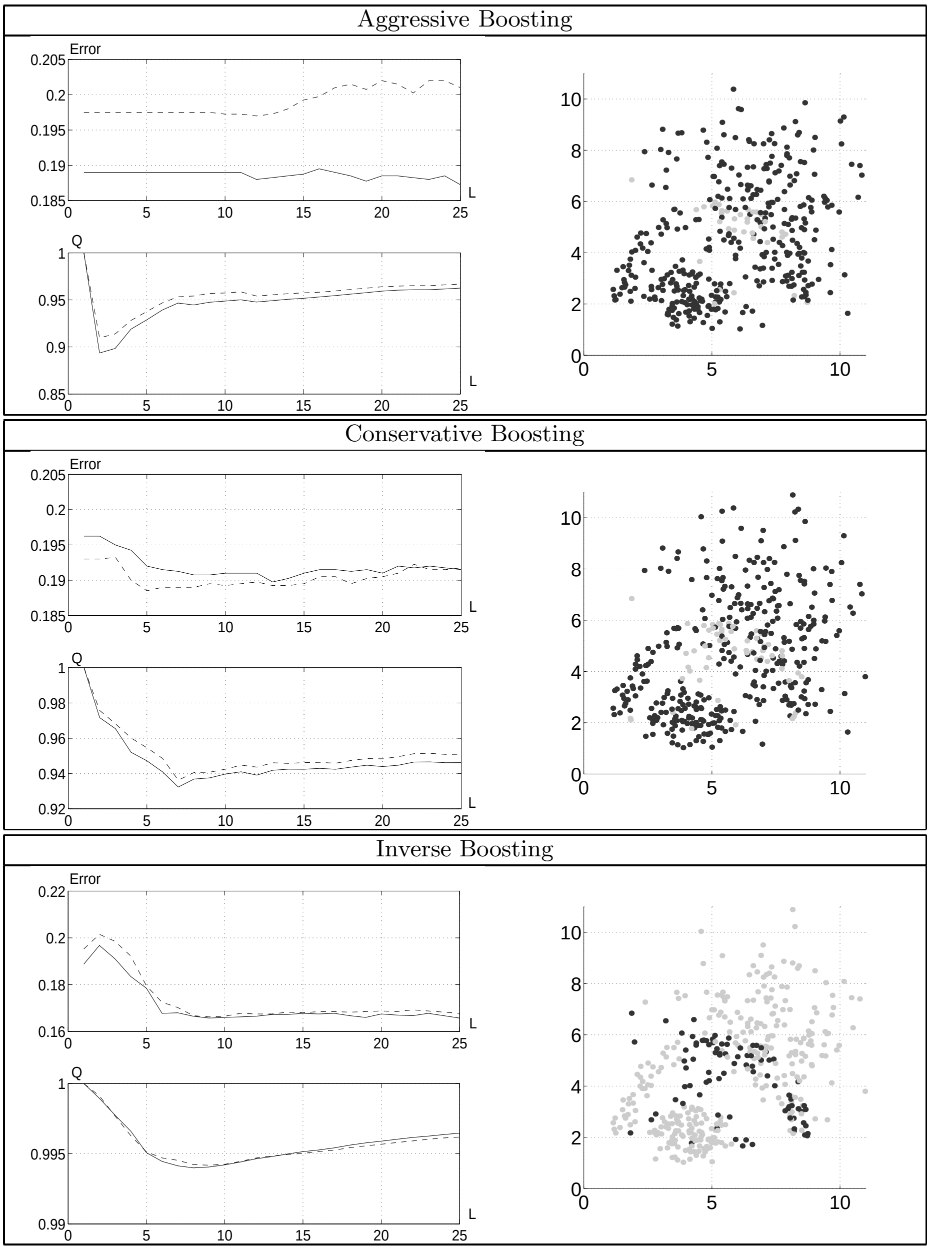


Fig. 2. Results from the three variants of ADAboost and the Cone-Torus data. On the left, for each variant, we show: *top plots*: The training error rate (solid line) and testing error rate (dashed line); *bottom plots*: The training Q(solid line) and the testing Q (dashed line). The corresponding point likelihoods are plotted on the right. Light gray points have highest weights (highest likelihood).

showing that there is a small L for which the classifiers are most diverse, and with L increasing the ensemble loses this diversity.

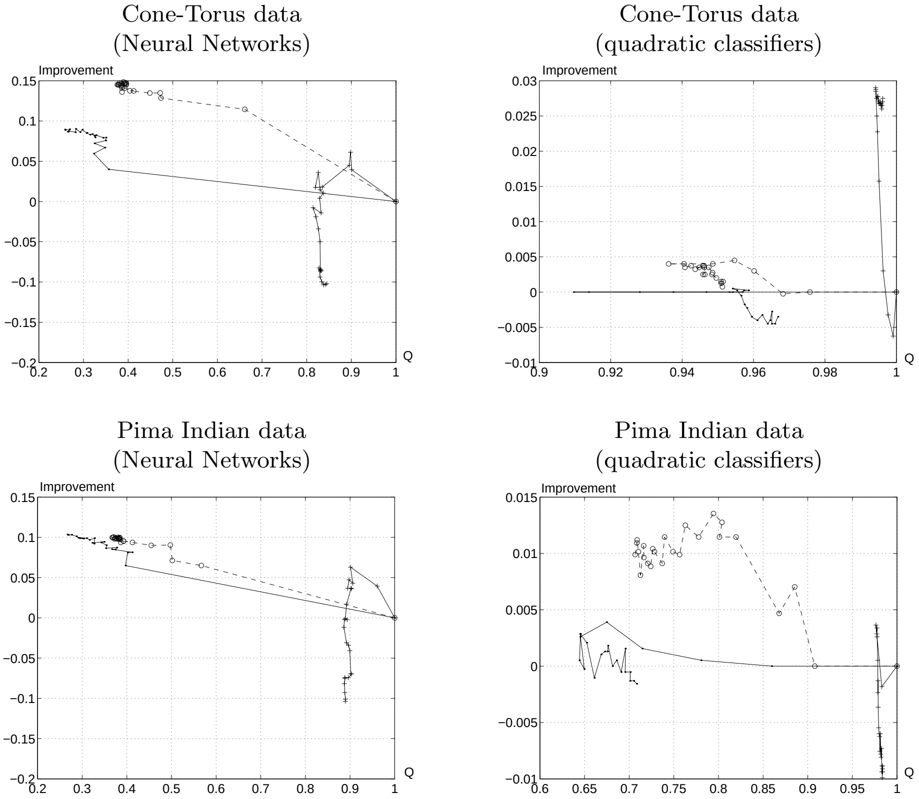
- For this particular example, “proper” Boosting was not the most successful ensemble building strategy. Inverse Boosting gave better results (lower error rates), although not much lower than the starting error rate.
- As could be expected, the Aggressive and the Conservative Boosting concentrate on the boundary points (see the scatterplots), and the Inverse Boosting does the opposite.
- Note the large difference between the Q values. Even though all training and testing curves had minima, judging by the absolute values of Q, the diversity for the Inverse Boosting is nonexistent. Still, the minimum, however shallow it is, indicates a reasonable place to stop the training.

It is curious to find out how the methods compare to each other in terms of both diversity and performance. We plotted the improvement on the single classifier (the starting classifier for $\mathcal{D}$) versus the diversity Q. To study the differences in the performances we used two basic classifier models: the quadratic discriminant classifier, and an MLP neural network with one hidden layer consisting of 15 nodes. For each classifier, the training was performed for 300 epochs using the fast backpropagation algorithm from the Matlab Neural Network Toolbox. Ten random splits of the data into halves were used for training and for testing, respectively, and the results were averaged. We used $L = 25$ as the final number of classifiers. The figures in the rest of this paper show results on unseen testing data only. Within this set-up, we have four combinations: 2 data sets × 2 base classifier models. The three Boosting methods for the four cases are plotted in Figure 3. The successive points for k from 1 to 25 are joined by lines. The y-axis in all figures show the testing accuracy minus the accuracy of the first classifier. Thus all ensembles started from Q=1, and zero improvement.

The plots prompt the following comments:

1. The patterns of performance are not consistent: there is no “best” Boosting variant amongst the three. Of course we can rate the performances noticing that the Inverse Boosting was only beneficial for the Cone-Torus data with quadratic classifiers where the other two methods were useless there. However, this looks more like a fluke than a serious finding. The improvement is not matched in the Pima Indian data plot for boosting quadratic classifiers. In fact, the performance declines after the first few “healthier” classifiers are added to the team, and purifying the training data further only harms the overall accuracy. From all three Boosting variants, perhaps the Conservative Boosting has the best overall performance, managing some improvement in all cases, notably better than the other two on the Pima Indian data plot with quadratic classifiers.

2. Looking at the scales of the two plots for the quadratic classifier and these for the MLP, there is a dramatic difference in the improvement on the single best classifier. While boosting quadratic classifiers leaves us with maximum 1.5 to 3 % improvement, when we combine neural network classifiers, the improvement goes up to 15 %. This confirms the results found by others that boosting makes



Cone-Torus data (Neural Networks)

Cone-Torus data (quadratic classifiers)

Pima Indian data (Neural Networks)

Pima Indian data (quadratic classifiers)

Fig. 3. Plots of improvement versus diversity Q for the two base classifier models and the two data sets. The solid line with the dots corresponds to the Aggressive Boosting, the dashed line with the circles corresponds to the Conservative boosting, and the solid line with the pluses corresponds to the Inverse Boosting.

sense for "capable" classifiers such as neural networks, whereby the possible overtraining is compensated for.

3. Diversity Q is not always a good indicator of the performance. For example, the lowest Q (highest diversity) will fail to detect the highest improvement for the Inverse Boosting (see Figure 3) in all cases except the Cone-Torus data and the quadratic base classifier. Even for that case Q is not too indicative. If we stopped at the lowest Q for the Pima Indian data and the quadratic classifiers, we would have missed the best improvement on all three Boosting methods. However, when the ensembles consist of neural networks, and the improvement is significant, stopping at the lowest Q will lead to the highest improvement both with the Aggressive Boosting and the Conservative Boosting. Notice also that the span of the diversity is much wider than for boosting quadratic classifiers. This indicates that while the relationship between diversity and accuracy might be blurred when Q spans a short interval of values, when a large improvement on

the accuracy is possible, the relationship between diversity and accuracy might become more prominent.

To examine our findings for different sizes of the training data we used the *Phoneme dataset* from UCI. Three training sizes were considered: small, $N = 80$, medium, $N = 350$, and large, $N = 1000$. Ten experiments were carried out with randomly dividing the data set into training and testing. The results are displayed on six Q-error plots in Figure 4 using the same line style as before.

The previous findings were confirmed and we also note that with the Phoneme data Aggressive Boosting gave the most diverse classifiers but Conservative Boosting managed to reach lower testing errors with less diverse classifiers. This suggests that Aggressive Boosting overemphasizes diversity which might result in ensembles with diverse but poor individual members. Conservative Boosting seemed to find a better compromise. The plots also show that Inverse Boosting leads the ensemble in the wrong direction of increasing the testing accuracy. The values of Q were approximately 1, indicating almost identical classifiers. Curiously, we did not find big differences for the different sample sizes with the NN classifiers. The patterns with the small data sets indicated that Aggressive and Conservative Boosting drive the testing error down for both classifier models whereas for larger data sets, the quadratic classifier behaves erratically. The reason for this is probably the fact that for small data sets, the quadratic classifier is no longer "stable". In other words, adding or removing a few data points will cause a sufficient change in the estimates of the covariance matrices to "destabilize" the quadratic classifier thereby making it suitable for boosting. The downside however is that such classifiers might not be accurate enough and therefore the total accuracy of the ensemble might suffer.

5 Conclusions

In this study we distinguish between three models of Boosting: Aggressive, Conservative and Inverse. We use an example of two data sets and two base classifier models to relate diversity in the ensemble and the improvement on the single classifier accuracy. Our results show that this relationship can be useful when the base classifier is flexible, leading to ensembles of high diversity albeit with possible overtraining of the individual members. Figure 2 suggests that the minimum Q identifies a sensible number of classifiers to include in the ensemble. However, paralysis was not induced in our experiments with the neural network classifiers, which are commonly accepted to be one of the more suitable models for Boosting. Therefore we were unable to confirm that Q identifies where paralysis begins. We also found that the Inverse Boosting quickly leads to a decline in the ensemble accuracy, emphasizing again the benefits of trying to produce diverse ensembles. The Conservative Boosting, which can be thought of as a softer alternative of the Aggressive Boosting exhibited better performance than the other two, and we therefore recommend it for practice.

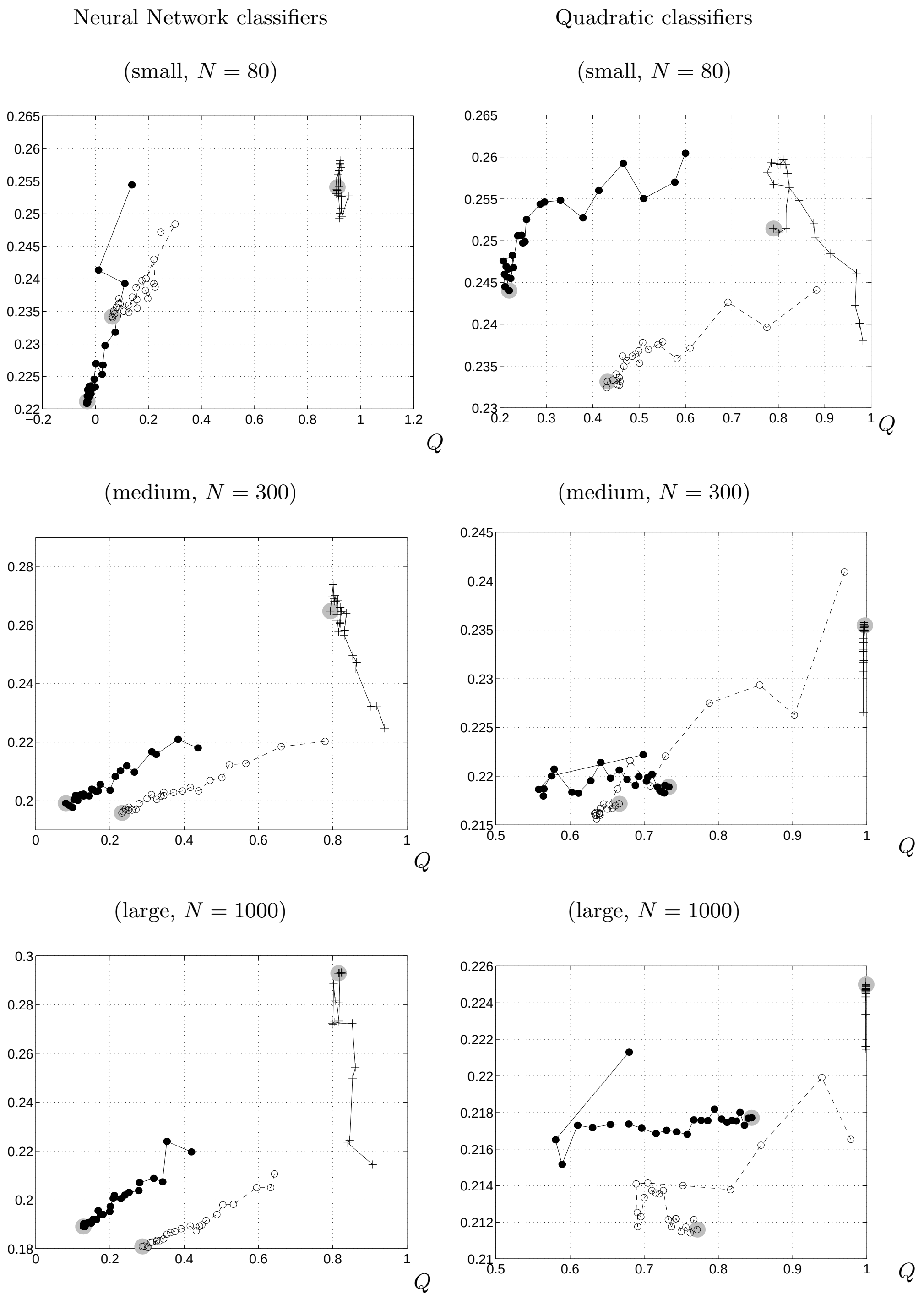


Fig. 4. Plots of error versus diversity Q for the two base classifier models and three sample sizes for the Phoneme data. The gray dot shows the stopping point.

References

1. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–142, 1999.
2. P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. Technical Report TCD-CS-2000-02, Department of Computer Science, Trinity College Dublin, 2000.
3. T.G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15, Cagliari, Italy, 2000. Springer.
4. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, NY, second edition, 2001.
5. Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
6. S. Hashem, B. Schmeiser, and Y. Yih. Optimal linear combinations of neural networks: an overview. In *IEEE International Conference on Neural Networks*, pages 1507–1512, Orlando, Florida, 1994.
7. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. MIT Press, Cambridge, MA, 1995.
8. L.I. Kuncheva. *Fuzzy Classifier Design*. Studies in Fuzziness and Soft Computing. Springer Verlag, Heidelberg, 2000.
9. L.I. Kuncheva and C.J. Whitaker. Ten measures of diversity in classifier ensembles: limits for two classifiers. In *Proc. IEE Workshop on Intelligent Sensor Processing*, pages 10/1–10/6, Birmingham, February 2001. IEE.
10. L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, and R.P.W. Duin. Is independence good for combining classifiers? In *Proc. 15th International Conference on Pattern Recognition*, volume 2, pages 169–171, Barcelona, Spain, 2000.
11. L. Lam. Classifier combinations: implementations and theoretical issues. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 78–86, Cagliari, Italy, 2000. Springer.
12. B.E. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science*, 8(3/4):373–383, 1996.
13. R.E. Schapire. Theoretical views of boosting. In *Proc. 4th European Conference on Computational Learning Theory*, pages 1–10, 1999.
14. C.A. Shipp and L.I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*. (accepted).
15. P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy*. W.H. Freeman & Co, 1973.
16. K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3/4):385–404, 1996.
17. J. Wickramaratna, S. Holden, and B. Buxton. Performance degradation in boosting. In J. Kittler and F. Roli, editors, *Proc. Second International Workshop on Multiple Classifier Systems*, volume 2096 of *Lecture Notes in Computer Science*, pages 11–21, Cambridge, UK, 2001. Springer-Verlag.
18. G.U. Yule. On the association of attributes in statistics. *Phil. Trans., A*, 194:257–319, 1900.