ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 123-128

AdaBoost and neural networks

T. Windeatt R.Ghaderi
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, U.K. GU2 5XH
T.Windeatt@ee.surrey.ac.uk R.Ghaderi@ee.surrey.ac.uk

Abstract

AdaBoost, a recent version of Boosting is known to improve the performance of decision trees in many classification problems, but in some cases it does not do as well as expected. There are also a few reports of its application to more complex classifiers such as neural networks. In this paper we decompose and modify this algorithm for use with RBF NNs, our methodology being based on the technique of combining multiple classifiers.

1 Introduction

A popular approach to improve generalisation is to combine unstable classifiers such as neural networks and decision trees which have low *bias*. Their problem is *variance*, which may be reduced by combining different classifiers(experts) that are produced by perturbing the training sets. Methods like *Bagging* [2], use random perturbation and simple voting so each expert is independent from the others (*parallel*). *I*n *Boosting* type methods , which are supposed to reduce both bias and variance[1], the training set of each expert has been filtered by previous expert (*sequential*). Some reports [1, 4, 10]indicate that great improvement is achieved by these techniques in decision trees upon many artificial and real data sets, but in few cases they do not do as well as expected. Investigations into these techniques are directed toward a few objectives. Some efforts are concerned with mathematical analysing to improve the re-sampling procedure [3, 6, 8, 11, 14]. Others such as [7, 13, 15] are about multi class problems. Using these algorithms with other classifiers [16, 5], or combining these techniques with other fusion methods are also fields of research [7]. In this paper we present different methods of combining to use them in AdaBoost.

2 Boosting and combining classifiers

Boosting was introduced by Schapire [12], as a method for boosting the performance of a weak learner. After some improvements by Freund [6], AdaBoost(Adaptive Boosting) was introduced by Schapire and Freund [8].

Although AdaBoost has evolved [8, 9, 14], we believe that an analysis from the viewpoint of general combining methodology [17] will help to modify it for complex classifiers such as neural networks. These experts are not “weak

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 123-128

learners" so designer should take care when adding complexity, by increasing the number of experts, not to damage regularisation and decrease efficiency.

Although in most cases combining methodologies have been heuristic, there are a few essential requirements in any application, and different strategies to satisfy them [17]. In AdaBoost these requirements are satisfied as follows:

For *Producing independent classifiers or correlation reduction between experts* All experts in AdaBoost use identical feature set and identical kind of learning machine but each one uses its own training set (picked up with a distribution calculated in each iteration) .

For *Combining mathematical framework* AdaBoost uses a hard-level combining with a weighted voting, in which the vote (decision) of each expert has fixed weighting factor based on its performance over training set.

Consider the space of composite classifiers$\{h_c = \sum_{t=1}^{T} \alpha_t.h_t | \sum_{t=1}^{T} \alpha_t = 1\}$, where h_c is composite hypothesis, h_t is the hypothesis trained on samples from a certain distribution D_t in t-th iteration, and α_t is the related weighting factor. AdaBoost is expected to find a near optimum solution in this space ($h_c^* = \sum_{t=1}^{T} \alpha_t^*.h_t^*$) *asymptotically*, by changing the distribution of training set based on correct classification, and choosing the α_t as a logarithmic function of average error of h_t. For many applications *asymptotically* means a few hundred of iterations, so a few points should be considered :

- In many cases particularly with complex learners such as neural networks, increasing the number of iterations reduces efficiency and is not so practical, but with fewer iterations the logarithmic weighting factors do not seem to be optimal. Other combining frameworks may provide better performance when efficiency is a consideration.

- If we assume $(\{\alpha_t^*\}_{t=1}^{T})$ as the optimal set, it means that over the whole of the test set $(\{(x_i, y_i)\}_{i=1}^{N})$ we have:
  $\sum_{i=1}^{N} \mathcal{I}(\sum_{t=1}^{T} \alpha_t^*.h_t^*(x_i), y_i) = Max \sum_{i=1}^{N} \mathcal{I}(\sum_{t=1}^{T} \alpha_t.h_t^*(x_i), y_i)$
  where $\mathcal{I}(x_1, x_2) = \begin{cases} 1 & if x_1 = x_2 \\ 0 & otherwise \end{cases}$. In this viewpoint, α_t is independent of x_i, however for different partitions of test set one may find other sets with better performance just over that partition. In other words, we may consider α_t as a function of x_i. Estimating such a function, we may improve performance. *Dynamic weighting factors* is such an approach in which another neural network has been trained to estimate a suitable weighting factor for each pattern. It is referred to as an Oracle and its design technique is given in[17].

3 Experiments

We provide experiments to explore the above points. In our approach, RBF NNs with fixed selected centres and fixed variance are chosen as learners. Another

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 123-128

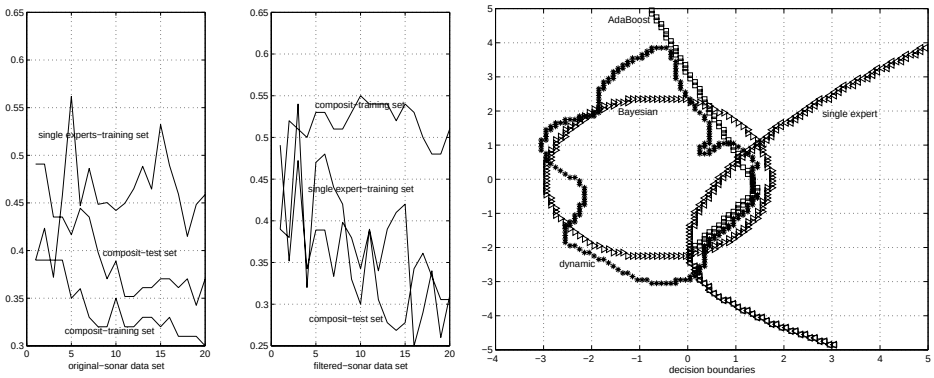

Figure 1: Error rate (%)of experts and AdaBoost on training and test sets(left)
Figure 2: Decision boundaries of Bayesian AdaBoost and Dynamic weighs(right)

RBF with 25 centres is used as Oracle in Dynamic weighting which is compared with some popular combining frameworks as well as AdaBoost.

Since the average rate of correct classification is not helpful in investigating the behaviour of classifiers in different parts of test set, we use an artificial benchmark which has a simple theoretical solution of the best possible classifier(*Bayesian*). With two dimensional inputs, it is suitable for visualisation of decision boundaries of classifiers, which helps us to compare the behaviour of classifiers in different parts of feature space. The problem consists of two groups of two dimensional random vectors with Gaussian distribution (mean $[0, 0]$, var 1 and mean $[2, 0]$, var 4)[18]. We present the results of two real data sets, diabetic and sonar[1], as well.

In AdaBoost, changing the D_t which is calculated with respect to error rate of h_{t-1} on training set, will change the h_t. If the test set of individual experts is the same set as its training(filtered set), error rate of individual experts decreases rapidly, but it does not mean better generalisation. Having a look at figures 1 and table 1 [sonar] to compare the use of original and filtered data set for *updating* step, we conclude that filtered adds more complexity so we can see that for simpler experts this approach has good results; but for more complex learners, using original data set is better.

Figure 2 [artificial] shows decision boundaries of first expert, Bayesian classifier as reference, AdaBoost and Dynamic weighting factors combiner, when RBF with 5 node is chosen as learning machine and 20 experts are produced by AdaBoost. It is shown that although both AdaBoost and dynamic weighting combiner successfully mimic the Bayesian classifiers decision (central circle) specially where the number of patterns is high, dynamic weighting is closer to Bayesian, this conclusion is confirmed by the result in table 2.

Results of AdaBoost(H-log), soft-level weighted averaging(S-log), order stat-

[1]thanks to UCI Machine Learning reposity

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 123-128

# centres	single	H-log	S-log	median	min/max	dynamic
5 -fil.	50.93	61.62	63.98	59.17	66.25	63.52
orig.		61.39	63.70	61.17	65.60	83.06
20 -fil.	54.63	64.69	65.12	66.79	66.17	67.55
orig.		78.49	82.08	79.96	80.17	88.51
50 -fil.	78.70	69.57	73.21	74.51	74.63	76.11
orig.		88.06	89.07	88.15	83.33	89.54
80 -fil	87.96	77.41	81.57	82.5	81.76	83.22
orig.		89.12	89.57	89.74	82.41	89.81

Table 1: classification rate (%) on sonar database using filtered and original data in updating on 20 independent run and 10 experts -100 sample for training and 106 sample for test set fil = filtered and orig=original

# centres	single	H-log	S-log	median	min/max	dynamic
2 -fil.	61.33	74.04	72.06	61.99	60.24	73.48
orig.		73.44	72.76	60.44	63.33	76.98
5 -fil.	69.11	72.52	74.27	72.63	72.24	73.81
orig.		77.66	76.53	72.22	66.42	78.22
10 -fil.	78.89	73.10	74.61	70.16	68.77	72.9
orig		78.41	79.17	64.59	65.2	78.82

Table 2: classification rate (%) on artificial database 20 independent run and 10 experts -100 sample for training and 900 sample for test set

#centres	single	H-log	S-log	median	min/max	dynamic
5 -fil.	65.66	66.92	67.74	68.12	69.26	70.39
orig.		70.0	70.2	71.56	70.38	72.32
20 -fil.	73.23	68.8	68.18	66.1	69.99	67.61
orig.		72.17	73.132	71.06	72.12	72.63
30-fil.	73.23	73.18	73.12	72.39	71.60	73.11
orig.		71.72	72.6	76.68	72.35	71.84

Table 3: classification rate (%) on diabetic database 20 independent run and 10 experts -570 sample for training and 214 sample for test set

	single	H-Boost	S-Boost	median	min/max	dynamic
mean	66.41	67.27	67.9	68.14	66.24	87.67
std	8.46	7.25	7.35	7.52	8.86	8.03

Table 4: results of classification (%) on diabetic database 10 independent run and 570 randomly chosen sample are used as training set

istics(mean, min/max) and dynamic weighting factors combiners on above data sets, presented in tables 1,2,3 show that dynamic weighting factors has better performance in comparison with other frameworks, and although for more complex NNs the gain of all combining methods is reduced, the sensitivity of this method is less.

In diabetic data set,an instance of difficult problem which involves noise and highly overlapped classes, it seems that AdaBoost can not provide a set of suitable experts because of many local minima in search space [10, 11]. In another approach to produce suitable individual experts we have used traditional type of Boosting for this problem. Here 90 samples are used to train first RBF with 25 centres, the second one trained on a data set with 50% errors and 50% from correct cases and the third one is trained on samples for which first two experts disagree. Different combiners are compared with the case that all training patterns are used for a single classifier. A MLP (25 hidden nodes in one hidden layer trained with LM algorithm)is used as oracle in this experiment, and results presented in table 4 indicate a great improvement in classification rate with dynamic weighting.

4 Conclusion

In this work we address dynamic weighting factors which is a flexible technique with good performance for combining classifiers, particularly when more complex learning machines such as NNs, are used. Experimental results on correct classification rate and visualising decision boundaries show better performance in comparison with logarithmic weighting factors of AdaBoost and other combining frameworks.

Acknowledgement Reza Ghaderi is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his Ph.D studies.

References

[1] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithm: Bagging, boosting, and variants. Boston. manufactured in the Netherlands, 1998. Machine Learning, Kluwer Academic Publisher.

[2] L. Breiman. Bagging predictors. volume 24, pages 123–140, Boston. manufactured in the Netherlands, Aug. 1996. Machine Learning, Kluwer Academic Publisher.

[3] L. Breiman. Bias, variance, and arcing classifiers. Technical Report 460, statistic dept. university of California, California,Berkeley CA. 94720, 1998.

[4] H. Drucker and C. Cortes. Boosting decision trees. pages 479–485. Advances in Neural Information Processing systems 8', 1996.

[5] H. Drucker, R.E. Schapire, and P. Simard. Improving performance in neural networks using a boosting algorithm. volume 5, pages 42–49. Advances in Neural Information Processing systems, Morgan Kaufmann, 1993.

[6] Y. Freund. Boosting a weak learning algorithm by majority. pages 202–216. Third workshop on computational learning theory, Morgan-Kaufman, 1995.

[7] Y. Freund, R. Lyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. 15th International conf. on Machine Learning, 1998.

[8] Y. Freund and R.E. Schapire. A decision-theoretic generalisation of on-line learning and application to boosting. *Journal of computer and system science*, 55:119–139, Aug. 1997.

[9] T. Onoda, K. Ratsch, and K.R. Muller. An asymptotic analysis of adaboost in the binary classification case. International conf. on Artificial Neural networks(ICANN'98), 1998.

[10] J.R. Quinlan. Bagging,boosting, and c4.5. volume 1, pages 725–730, Menlo park CA, USA, 1996. 13th National Conf. on Artificial Intelligence, AAAI.

[11] G Ratsch and T Onoda. Soft margin for adaboost. Technical Report 021, NeuroCOLT, Rudowe Chaussee 5, 12489 Berlin,Germany, August 1998.

[12] R. E. Schapire. The strength of weak learnability. Number 2 in 5, pages 197–227. Machine Learning, 1990.

[13] R.E. Schapire. Using output codes to boost multiclass learning problems. 14th International conf. on Machine Learning, 1997.

[14] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. pages 322–330. 14th International conf. on Machine Learning, Morgan Kaufman, 1997.

[15] R.E. Schapire and Y. Singer. Improved boosting algorithm using confidence-rated predictions. 11th annual conf. on computational learning theory, 1998.

[16] H. Schwenk and Y. Bengio. Adaptive boosting of neural networks for character recognition. volume 1327, pages 967–972. International conf. on Artificial Neural Networks(ICANN"97)Berlin, Springer, 1997.

[17] T. Windeatt and R. Ghaderi. Dynamic weighting factors for decision combining. pages 123–130, Great Malven, U.K., 1998. International Conf. on Data Fusion.

[18] P. Yee. Classification requirements involving backpropagation and rbf networks. Technical Report 249, Comm. Res. Lab. McMaster Univ., Ontario, 1992.