Attribute Selection Methods for Filtered Attribute Subspace based Bagging with Injected Randomness (FASBIR)

I. M. Whittley[1], A. J. Bagnall[1], L. Bull[2], M. Pettipher[3], M. Studley[2] and F. Tekiner[3]

[1] School of Computing Sciences, University of East Anglia, Norwich, England
[2] School of Computer Science, University of West of England, England
[3] Department of Computer Science, University of Manchester, England

Abstract. Filtered Attribute Subspace based Bagging with Injected Randomness (FASBIR) is a recently proposed algorithm for ensembles of k-nn classifiers [28]. FASBIR works by first performing a global filtering of attributes using information gain, then randomising the bagged ensemble with random subsets of the remaining attributes and random distance metrics. In this paper we propose two refinements of FASBIR and evaluate them on several very large data sets.

keywords: ensemble; nearest neighbour classifier

1 Introduction

As part of an EPSRC project developing data mining tools for super computers, we are examining the best ways of employing emsembles of classifiers for data sets with a large number of attributes and many cases. *Filtered Attribute Subspace based Bagging with Injected Randomness* (FASBIR) is one of several recently proposed algorithms for ensembles of k-nn classifiers [28]. FASBIR works by first performing a global filtering of attributes using information gain, then constructing a bagged ensemble with random subsets of the remaining attributes and diversified distance measures.

The main contributions of this paper are, firstly, to collate and format a set of large attribute many case data sets used in related research and, secondly, to propose refinements for FASBIR that make it more suitable for use with these large datasets. Other algorithms proposed for ensembles of k-nn classifiers include Multiple Feature Subsets (MFS) [5] and Locally Adaptive Metric Nearest Neighbour (ADAMENN) [10]. We believe that FASBIR is the most promising approach for high dimensional data with a large number of observations. MFS will perform poorly with a large number of attributes. ADAMENN requires several scans of the data to classify a new case and hence is not appropriate for data set with many records. FASBIR reduces the dimensionality with a simple filter and is fast to classify new cases. However, three features of FASBIR (as

described in [28]) can be improved upon for dealing with large data. Firstly, the filter is performed on the whole data prior to ensembling. For large training sets, this can prohibitively expensive, particularly if there is little overlap in data cases between the ensembles. Secondly, FASBIR has a large number of parameters and traditional wrapper approaches for setting these values are not feasible for large data. Thirdly, FASBIR combines the output of the classifiers using a simple voting scheme. In this paper, for consistency with [28] we assume there is a large overlap in the data samples between classifiers, hence there is little to gain from localised filtering. Instead, we propose randomising parameters across the ensemble and using a probability based voting scheme.

The remainder of this paper is structured as follows. In Section 2 we briefly review related work. In Section 3 we describe the refinements introduced to improve FASBIR. In Section 4 we describe the data sets used in the experimentation reported in Section 5 and discuss our next objectives in the conclusions 6.

2 Background

Nearest neighbour (NN) classifiers, first proposed in 1951 by Fix and Hodge [16], are very simple forms of non-parametric, lazy classifiers that have remained a popular exploratory data analysis technique. A NN classifier consists of a training data set and a distance metric. New cases are assigned the class of the closest case in the training set. A common extension is k nearest neighbour (k-NN) [1], which involves finding the k nearest cases then assigning the most common class of these cases to the new case. NN search forms a core activity in many fields, such as time series data mining [24], pattern recognition [3] and spacial and multimedia databases [26]. A collection of the important papers on k-NN classifiers was published in 1990 [13].

The most commonly recognised problems with k-nn classifiers are that, firstly, they are sensitive to redundant features and secondly, classifying new cases is relatively time consuming for large data sets [2].

A recently popular research theme which partially addresses these problems is methods for combining several k-NN classifiers through *ensembles*. Ensemble techniques amalgamate the predictions of several classifiers through a voting scheme to form a single estimated class membership for each test case. Ensemble techniques can be categorised as two types: Sequential methods which iteratively build classifiers on the same data set with different weight functions (examples include ADABOOST [17], ARC-x4 [8], ECOC [25] and LogitBoost [18]); and parallel methods that construct classifiers concurrently on possibly different data sets, such as Bagging [7], Random Forest [9] and Randomization [14].

In the paper first introducing the Bagging technique [7] it is shown that bootstrapping k-NN classifiers does not result in better accuracy than that obtained with the whole model. If the ensemble samples are found by exhaustively sampling without replacement, and each ensemble still finds the closest k in the subset, then it is trivial to recreate the whole data k-NN from the ensemble. However, it is also well known that k-NN is sensitive to variation in the at-

tribute set. Since the objective of designing ensembles is to increase diversity while maintaining accuracy, these factors have meant that the major focus of research into ensembles of k-NN classifiers has been on methods to select subsets of attributes. For example, Bay [5] evaluates the effect using a random subset of attributes, called Multiple Feature Subsets (MFS), for each Nearest Neighbour member of the ensemble. MFS is evaluated when used with sampling with replacement and sampling without replacement on 25 small datasets from the UCI Repository of Machine Learning Databases [6]. MFS was compared to NN, k-NN and greedy feature selection algorithms (forward selection and backward elimination) described in [2]. MFS combines the votes of the individual classifiers. The method described in [23] also involves random attribute splits, but differs from MFS in that it combines the nearest neighbours of each ensemble rather than the votes. Random subspaces have also been used in [19]. Zhou's FASBIR [28] first measures the information gain of each attribute, then removes all the attributes with information gain less than some threshold. Bootstrapping samples are formed from the resulting dataset, and each classifier is assigned a random subset of attributes and a randomized distance metric (injected randomness). The algorithm is evaluated on 20 small data sets from the UCI repository. Domeniconi and Yann describe the Locally Adaptive Metric Nearest Neighbour (ADAMENN) classifier in [10] and how it could be used for ensembles in [15]. ADAMENN produces a probability distribution over the attribute space based on the Chi-squared statistic. It produces this distribution for each new case in the test data. For every classifier in the ensemble the attribute distribution is sampled (both with and without replacement) to form a subset of fixed size. The ADAMENN ensemble is evaluated on five small data sets from [6].

Filtering has been shown recently to be as effective as more complex wrapper methods on a range problems with a large number of attributes [21]. Since there are obvious speed benefits for filters over wrappers we believe that FASBIR is the most promising k-nn ensemble approach for the data with a large number of attributes and many cases.

3 FASBIR

The FASBIR [28] algorithm is summarised in Figure 3. A Minkowsky distance metric for ordinal attributes and the Value Difference Metric for nominal attributes is used. The set of distance functions for FASBIR is a set of possible values for the power p of the Minkowsky/VDM measure, which in [28] is restricted to the set $C = \{1, 2, 3\}$. The other parameters are listed in Table 1.

In this paper we test two refinements of FASBIR. The first is an improvement to the prediction mechanism. The algorithm as described in [28] uses simple majority voting. Each member of the ensemble predicts the class and these votes are collected to determine the predicted class. Our basic refinement is to make each classifier produce a probability estimate for class membership. These probabilities are then combined. This allows the ensemble to retain more of the

Algorithm 1 The FASBIR Algorithm

Given training data set D with n cases and m attributes, **train**

1. Filter the attributes
   (a) Measure IG on each attribute. Let a be the average information gain over the f attributes.
   (b) Discard any attribute with IG less than $f \cdot a$, giving m' attributes.
2. For each of the t classifiers in the ensemble
   (a) sample with replacement $n' = r \cdot n$ data
   (b) sample without replacement $s * m'$ of the filtered attributes
   (c) select a random distance measure from a set of candidates C

For each new case to classify **test**

1. For each of the t classifiers in the ensemble
   (a) Find the k nearest neighbours with selected distance metric
   (b) Return the majority class of the neighbours
2. Classify case as the majority class of all the ensemble votes

Table 1. FASBIR Parameters

Parameter	Meaning	Setting in [28]
f	proportion of attributes to filter	0.3
s	proportion of attributes to randomly select	0.25
r	proportion of data to sample	1
t	number of classifiers	20,40,60,80,100
k	number of neighbours	1,3,5,7,9

discriminatory power in the constituent classifiers. The second refinement is a generalisation of the parameter space. The parameter values for f and s are fairly arbitrary. For problems with redundant attributes, fixed cut off values may be sufficient to capture the important attributes. However, if there is multicollinearity and deceptive/partially useful attributes, the filter may retain or remove fields of use. One of the driving forces in this algorithm's design is the need to diversify the classifiers. Hence, rather than have a fixed cut off value f and s, we randomised the filter value for each classifier in the ensemble. These refinements are assessed in Section 5.

4 Data Sets

We have collected 18 data sets, 9 from attribute and model selection competitions ([22, 11, 20, 12]), 2 standard sets from the UCI repository, 2 simulated sets and 5 new problems provided by contributors to our EPSRC project [27]. Summary information on the datasets is given in Table 2. Further information on all the data is available from [4]. We have included the very small Glass problem to provide validation that our results are comparable to those obtained in

the FASBIR paper. All continuous attributes are normalised using a standard normal transformation.

Table 2. Data Set Summary

Source	Name	Size(KB)	CASES	Atts	Data	Ordinal	Nominal	Classes
NIPS2003	Madelon	5,085	2600	500	1300000	500	0	2
PASCAL2004	Catalysis	2,660	1173	617	723741	617	0	2
WCCI2006	Hiva	12,155	3845	1617	6217365	1617	0	2
WCCI2006	Gina	7,554	3153	970	3058410	970	0	2
WCCI2006	Sylva	6,531	13086	216	2826576	216	0	2
WCCI2006	Ada	432	4147	48	199056	48	0	2
IJCNN2006	Temp	8,159	7177	106	760762	106	0	2
IJCNN2006	Rain	8,159	7031	106	745286	106	0	2
IJCNN2006	SO2	4,532	15304	27	413208	27	0	2
UCL	Adult	3,882	32561	14	455854	8	6	2
UCL	Glass	12	214	9	1926	9	0	7
Commercial	ProductW	71,873	715028	43	30746204	4	39	6
Commercial	ProductX	60,968	590943	44	26001492	4	40	6
Commercial	ProductY	34,283	339312	43	14590416	4	39	6
Commercial	ProductZ	23,304	224693	44	9886492	4	40	6
Hunt	Obesity	2,240	12429	89	1106181	49	40	4
Simulated	SGR	18,348	100000	100	10000000	100	0	2
Simulated	SGC	18,348	100000	100	10000000	100	0	2

5 Results

These experiments test the effectiveness of FASBIR with and without refinements on the 18 data sets described in Section 4. We compare FASBIR to a simple naive Bayes classifier and linear discriminant analysis, both of which use all the features, and to MFS. FASBIR Vote, is the algorithm proposed in [28]. FASBIR Prob uses probability distribution voting. FASBIR Random uses probability voting and has randomised parameters.

Table 3 gives the testing accuracy with a 10-fold cross validation on the 18 data sets. The experiments reported in Table 3 were performed with $k = 7$. Further testing not reported showed that a very similar pattern of results was produced with k values from 1 to 9. Looking at the balanced error rate rather than the accuracy also led to similar conclusions, so these statistics are omitted for brevity.

The first observation we can make from these results is that, with the exception of the two simulated data sets, FASBIR Random is always more accurate than both Naive Bayes (NB) and Discriminant Analysis (DA). The other FASBIR versions are generally more accurate than NB and DA. This is simply a demonstration the value of feature selection in high dimensional feature spaces, and serves as a basic sanity check.

Table 3. Test accuracy averaged over 10 folds. The highest figure is in bold

	Naive Bayes	DA	MFS	FAS Vote	FASBIR Prob	FASBIR Random
Madelon	59.15%	54.88%	59.38%	58.69%	57.88%	**59.81%**
Catalysis	67.26%	60.02%	69.39%	**70.84%**	69.56%	69.64%
Hiva	43.38%		96.70%	96.67%	96.67%	**96.70%**
Gina	75.33%	82.24%	83.45%	94.45%	**94.83%**	94.61%
Sylva	95.77%	98.67%	97.91%	99.30%	**99.34%**	99.29%
Ada	51.31%	84.35%	83.53%	84.50%	84.57%	**84.71%**
Temp	89.94%	92.82%	92.60%	93.26%	**93.34%**	93.23%
Rain	75.81%	78.27%	**79.21%**	79.12%	79.05%	79.04%
SO2	80.83%	87.11%	87.15%	87.49%	87.44%	**87.50%**
Adult	83.35%	81.00%	83.78%	84.60%	**85.35%**	85.06%
Glass	15.37%	45.26%	71.08%	72.08%	73.44%	**74.85%**
ProductW	41.57%	23.61%	43.25%	42.48%	**44.71%**	44.59%
ProductX	45.67%	27.25%	45.96%	48.16%	**49.99%**	49.27%
ProductY	51.85%	22.31%	55.13%	55.70%	55.63%	**55.70%**
ProductZ	48.82%	29.14%	51.34%	52.08%	52.49%	**52.51%**
Obesity	44.36%	54.40%	54.67%	54.59%	**54.93%**	54.45%
SGR	64.99%	**98.21%**	67.95%	70.92%	71.02%	71.19%
SGC	**86.73%**	86.69%	85.91%	86.16%	86.35%	86.29%

The second observation we can make from these results is that filtering provides a benefit for ensembles of k-nn. Each FASBIR implementation has greater accuracy than MFS on 15 of the 18 data sets. If we view the data as a paired sample, we can reject the hypothesis that the difference in accuracy is zero using a sign test, a Wilcoxon sign rank test and a t-test, even after removing the outlier Gina. With Rand Fasbir and MFS, there is a significant difference between the classifiers on 12 out of the 18 data, as measured by McNemar's test. There is no significant difference on the other 6 data sets. These results corroborate the findings of [28].

Thirdly, we observe that there is a definite advantage in combining the probability estimates of the individual classifiers in the ensemble rather than combining votes. If we compare majority voting FASBIR and FASBIR with probability voting (columns 5 and 6 in Table 3), we see that probability voting is better on 12 out of the 18 data sets. This difference is also observable if we look at the difference between MFS with and without voting and FASBIR random with and without voting. For all three pairs, there is a significant difference between the classifiers on 6 out of the 18 data, and no significant difference on the remaining sets. This indicates there is no penalty for using the probabilities, and on many occasions a significant improvement can be achieved.

Fourthly, randomising the filter and selection parameters has no significant any effect on accuracy. Randomisation significantly improves performance on Madelon and SGC, and this results demonstrates the potential benefit of avoiding a fixed filter value. Both Madelon and SGC have correlated attributes. This multicollinearity is exactly the type of situation where an arbitrary filter may

remove relevant attributes. The benefits are small and our experiments do not conclusively support the use of randomised parameters across the ensemble. However, because of the benefits of reducing the parameter space and increased robustness, we believe it is a sensible approach.

6 Conclusions and Future Directions

Very large, many attribute data sets offer a unique type of challenge that is being faced by data mining practitioners with increasing frequency. Many approaches to attribute selection are simply not feasible with such massive data. Attribute filters are a simple and effective method that have been effective with this kind of data. In this paper we have collated several disparate sources of very large, many attribute data sets, including five never used before. We have proposed some minor modifications of a recently published filtering algorithm using in conjunction with k-nn, and demonstrated that filtering improves performance on the majority of these large data.

The next stage of this work will be to collect more data, to experiment with alternative distance metrics and evaluate alternative greedy randomised atribute selection algorithms.

Acknowledgement

This research was funded by the EPSRC under grant GR/T18479/01, GR/T18455/01 and GR/T/18462/01.

References

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.
2. D.W. Aha and R.L. Bankert. Feature selection for case-based classification of cloud types: an experimental comparison. In *Proc. AAAI 94*, pages 106–112, 1994.
3. S. N. Srihari B. Zhang. Fast k-nearest neighbor classification using cluster-based trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(4):525–528, 2002.
4. A. Bagnall. Large data sets for variable and feature selection. `http://www.cmp.uea.ac.uk/ ajb/SCDM/AttributeSelection.html`, 2006.
5. S. D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 37–45, 1998.
6. C.L. Blake and C.J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.
7. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
8. L. Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–849, 1998.
9. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
10. D. Gunopulos C. Domeniconi, J. Peng. Locally adaptive metric nearest-neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002.

11. J Q. Candela, C. Rasmussen, and Y. Bengio. Evaluating predictive uncertainty challenge, presented at NIPS 2004 workshop on calibration and probabilistic prediction in machine learning. `http://predict.kyb.tuebingen.mpg.de/`, 2004.
12. G. Cawley. Predictive uncertainty in environmental modelling competition, special session at ijcnn-2006. `http://clopinet.com/isabelle/Projects/modelselect/`, 2006.
13. B. Dasarathy. *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press, 1990.
14. T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
15. C. Domeniconi and B. Yan. On error correlation and accuracy of nearest neighbor ensemble classifiers. In *the SIAM International Conference on Data Mining (SDM 2005)*, 2005.
16. E. Fix and J. L Hodges. Discriminatory analysis, non-parametric discrimination: consistency properties. Technical report, USAF School of aviation and medicine, Randolph Field, 1951.
17. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conference on Computational Learning Theory*, pages 23–37, 1995.
18. J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:337–407, 2000.
19. R. Sabourin G. Tremblay. Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. In *17th International Conference on Pattern Recognition (ICPR'04)*, 2004.
20. I. Guyon. Model selection workshop, part of the IEEE congress on computational intelligence. `http://clopinet.com/isabelle/Projects/modelselect/`, 2006.
21. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.
22. I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. NIPS 2003 workshop on feature extraction. `http://www.nipsfsc.ecs.soton.ac.uk/`, 2003.
23. T. K. Ho. Nearest neighbors in random subspaces. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 640–648, 1998.
24. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.
25. E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 313–321, 1995.
26. T. Seidl and H. Kriegel. Efficient user-adaptable similarity search in large multimedia databases. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 506–515, 1997.
27. F. Tekiner. Super computer data mining project. `http://www.mc.manchester.ac.uk/scdm/toolkit`, 2006.
28. Z.-H. Zhou and Y. Yu. Ensembling local learners through multi-modal perturbation. *IEEE Transactions on systems, man, and cybernetics - part B*, In press - 2005.