PERGAMON

Pattern Recognition 36 (2003) 1291 – 1302

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets

Robert Bryll[a,*], Ricardo Gutierrez-Osuna[b], Francis Quek[a]

[a] Vision Interfaces & Systems Laboratory (VISLab), CSE Department, Wright State University, 303 Russ Engineering Center,
3640 Colonel Glenn Highway, Dayton, OH 45435-0001, USA
[b] Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA

Received 2 November 2001; accepted 12 June 2002

Abstract

We present attribute bagging (AB), a technique for improving the accuracy and stability of classifier ensembles induced using random subsets of features. AB is a wrapper method that can be used with any learning algorithm. It establishes an appropriate attribute subset size and then randomly selects subsets of features, creating projections of the training set on which the ensemble classifiers are built. The induced classifiers are then used for voting. This article compares the performance of our AB method with bagging and other algorithms on a hand-pose recognition dataset. It is shown that AB gives consistently better results than bagging, both in accuracy and stability. The performance of ensemble voting in bagging and the AB method as a function of the attribute subset size and the number of voters for both weighted and unweighted voting is tested and discussed. We also demonstrate that ranking the attribute subsets by their classification accuracy and voting using only the best subsets further improves the resulting performance of the ensemble.

Keywords: Ensemble learning; Classifier ensembles; Voting; Feature subset selection; Bagging; Attribute bagging; Hand-pose recognition

1. Introduction

Combining outputs from multiple classifiers, known as ensemble learning, is one of the standard and most important techniques for improving classification accuracy in machine learning [1,2]. Among these, bagging and boosting [3,4] are the most popular methods of ensemble learning. In bagging [3], the training set is randomly sampled k times with replacement, producing k training sets with sizes equal to the original training set. Since the original set is sampled with replacement, some training instances are repeated in the new training sets, and some are not present at all. The obtained sets are used to train classifiers (e.g. decision trees), giving k different predictors, which can be used to classify new data. The classification for each data instance is obtained by equal weight voting on all k predictors. Voting gives a significant improvement in classification accuracy and stability. Boosting, on the other hand, induces the ensemble of classifiers by adaptively changing the distribution of the training set based on the accuracy of the previously created classifiers and uses a measure of classifier performance to weight the selection of training examples and the voting.

Theoretical and empirical results [5,6] suggest that combining classifiers gives optimal improvements in accuracy if the classifiers are "independent", i.e. they are not correlated with one another. According to Ref. [5], the most effective method of achieving such independence is by training the members of an ensemble on qualitatively different feature (sub)sets. In other words, attribute partitioning methods are capable of performance superior to data partitioning methods (e.g. bagging and boosting) in ensemble learning.

There is a growing number of publications that investigate performance of classifier ensembles trained using attribute

[*] Corresponding author. Tel.: +1-937-775-3930; fax: +1-937-775-5133.
E-mail address: rbryll@cs.wright.edu (R. Bryll).
URL: http://vislab.cs.wright.edu/∼rbryll

0031-3203/03/$30.00 © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
PII: S0031-3203(02)00121-8

subsets. The issue of selecting optimal subsets of relevant features plays an important role in this research. The feature selection problem [7] can be viewed as a multi-criterion optimization problem that can be solved using various heuristic search methods, such as greedy approaches [8] or optimization techniques such as genetic algorithms [9]. John et al. [8] propose a two-degree model of feature relevance and discuss various approaches to the feature subset selection problem. Their paper advocates for the *wrapper model* of feature subset selection instead of the *filter model*. In the wrapper model, the induction algorithm (the classifier) is used to evaluate the feature subsets, whereas in the filter model the evaluation of subsets is based on measures of information content (e.g. inter-class distance, statistical dependence) that are independent of the particular induction algorithm [10]. An example of the filter model can be found in [10], where the authors train N classifiers for an N-class problem and the subset of features used to train each classifier is selected according to their correlation with the corresponding class. The disadvantage of the filter model is that it does not utilize the bias of the particular induction algorithm to select a fine-tuned feature set. In addition, it has a tendency to select large feature subsets since filter measures are generally monotonic [11]. Therefore, evaluating feature subsets using the classifiers themselves can be advantageous.

Randomized attribute subsets are often used to “inject randomness” [2] into the classifiers in order to increase their independence in the ensemble. In Ref. [12] the sets of attributes available to a decision tree induction algorithm at each node are randomly perturbed to achieve better classifier independence. The authors call this approach *Stochastic Attribute Selection Committees* (SASC). In Ref. [13] the authors combine SASC with boosting to achieve accuracy higher than either boosting or SASC alone. The problem with the methods presented in Refs. [12,13] is that they require modifications to the decision tree induction algorithm (C4.5 in this case).

The power of randomizing subsets of features in ensemble learning has also been demonstrated in Ref. [14], where the accuracy of nearest-neighbor-classifier ensembles was improved despite the fact that the performance of these ensembles is not improved by standard methods such as bagging or error correcting output coding [15]. The improvements are even more impressive because they were achieved without any accuracy ranking of attribute subsets used in voting (all random subsets were used).

This paper describes an approach to improving the results of voting on ensembles of classifiers induced using attribute subsets. Our method, called attribute bagging (AB) due to its conceptual similarity to Breiman’s bagging [3,4], combines the “wrapper” feature subset evaluation approach described in Ref. [8] with subset randomization methods [12–14]. We propose a framework that maximizes the accuracy of the ensemble of classifiers built on attribute subsets by

- Finding the appropriate size of the attribute subset and
- improving the voting performance of the ensemble.

The voting performance of an ensemble can be improved by using only the subsets of features that give best classification accuracy on training data or on a validation set. Our algorithm uses two processing stages to address the aforementioned issues.

First, an appropriate attribute subset size m is found by testing classification accuracy of variously sized random subsets of attributes. This “optimal” attribute subset size will obviously be problem dependent and, therefore, will be affected by the redundancy and correlations between features extracted for a given dataset.

In the second stage, classification accuracy of randomly selected m-attribute subsets is evaluated by the induction algorithm and only the classifiers constructed on the highest ranking subsets participate in the ensemble voting. In contrast with Refs. [12,13], our method can work with any classifiers and does not require modifications to the induction algorithms.

The feature subsets that we choose for voting are not specially selected to be qualitatively different from one another. We do not perform any test (such as computing correlations between features) to ensure independence of feature subsets. Instead, we rely on the randomness of the selection and the number of voters to yield sufficient number of qualitatively different classifiers. As evidenced by our experiments, only the number of voters higher than 15–20 (for our particular dataset) yields ensemble classifiers that are highly accurate and stable. The reason for this is that only if the number of voters is “large enough” does the random process of attribute selection yield sufficient number of qualitatively different classifiers that ensure high accuracy and stability of the ensemble.

This article compares the performance of our AB method with bagging and single-classifier algorithms such as C4.5, RIEVL and OC1 on a hand-pose recognition dataset [16]. We also present extensive tests and discussions of the accuracy and stability of ensemble voting in the AB method for a wide range of attribute subset sizes and varying number of voters in both weighted and unweighted voting. Our experiments demonstrate that AB gives consistently better results than bagging. In addition, the best achieved accuracy exceeded that of the RIEVL algorithm presented in Ref. [16]. Also, our tests show that the two-stage approach, in which the appropriate attribute subset size is established first, and only the best (most accurate) random attribute subsets are used in voting, gives results that are superior to those obtained with unranked random subsets of attributes as in Ref. [14].

2. Advantages of using attribute subsets

As outlined in Ref. [6], the advantages of using attribute subsets in ensemble learning include: (1) reduction of dimensionality in the data, which lessens the impact of the "curse of dimensionality"; (2) reduced correlation among the classifiers by training them on different features and, as a result, (3) improvement in the classification performance of the ensemble.

Moreover, since selecting the attribute subsets is equivalent to creating projections of the training set, the following two advantages of using attribute subsets, as opposed to data instance subsets, can be added:

(1) Projections are smaller in size than the original training set. As a result, the amount of data transferred and duplicated during the creation of the training sets is smaller than in bagging. The reduced size of the dataset results in faster induction of classifiers.
(2) All classes appearing in the original training set are always represented in the training sets created by projection, which guarantees that each predictor used in voting is built with full class representation. In bagging, some of the classes present in the original training set may be underrepresented or even absent from some of the derived training sets, which negatively affects predictors constructed from those sets.

3. Experimental setup

3.1. Learning algorithm and experimental code

In our experiments we use the OC1 decision tree system developed by Murthy et al. [17]. Although OC1 can produce splits oblique to the parameter space axes while inducing decision trees, we use it in the axis parallel mode, since in this mode building decision trees is much faster for datasets with many arguments. All the experiments reported in this article use OC1's default pruning parameters.

The experimental code was written in C++ on an IRIX 6.2 platform. It performs all necessary data manipulation (selecting training sets in bagging, selecting training and test sets for holdouts, etc.), invokes OC1's *mktree* program to induce predictors, and then carries out predictor evaluation and voting. In most cases, the above-mentioned operations are executed multiple times, and the experimental software computes averages and standard deviations of the predictor accuracies. Our experiments were performed on a 4-processor SGI Onyx workstation.

3.2. Dataset

In this article, we use the hand-pose/configuration dataset presented in Ref. [16], created for the purpose of vision-based hand-pose recognition. The database contains 20 hand configurations (20 classes), 29 attributes (including the target), 934 instances for training, and 908 instances in the test set. All attributes have been discretized into seven possible values. The attributes, which are summarized in Table 1, were computed for the hand area after the image was segmented based on color information. Fig. 1 shows a sample of six hand configurations in the database.

In most of our experiments we have used the same training and test set partitions of Ref. [16] in order to compare the performance of AB with that of the RIEVL algorithm and other learning systems listed in that article, such as C4.5 and ID3. In addition, we performed multiple holdout runs on the entire dataset to establish a comparison between AB and bagging.

4. Experimental results

4.1. Finding an appropriate attribute subset size

To determine an appropriate dimensionality *n* for each classifier in the ensemble, we evaluated the performance of ensembles containing 25 classifiers built on attribute subsets ranging from 1 to 25 features. As mentioned earlier, we used the same training and test partitions reported in Ref. [16]. First, we randomly selected 25 attribute subsets of the

Table 1
Attributes of the hand-pose dataset extracted in Ref. [16]

Attribute	Attribute description	Method of computation
X1	Hand bounding box aspect ratio	Rows/cols
X2	Hand bounding box area	Rows*cols
X3	Area of hand in the bounding box	
X4		X3/X2
X5–X6	Hand centroid coordinates	
X7–X13	Hand central moments	CM11, CM02, CM20, CM12, CM21, CM03, CM30
X14	Hand orientation	arctan(2*CM11/(CM20-CM02))
X15–X21	Normalized central moments	NM11, NM02, NM20, NM12, NM21, NM03, NM30
X22–X28	Rotational invariants	RI0, RI1, RI2, RI3, RI4, RI5, RI6

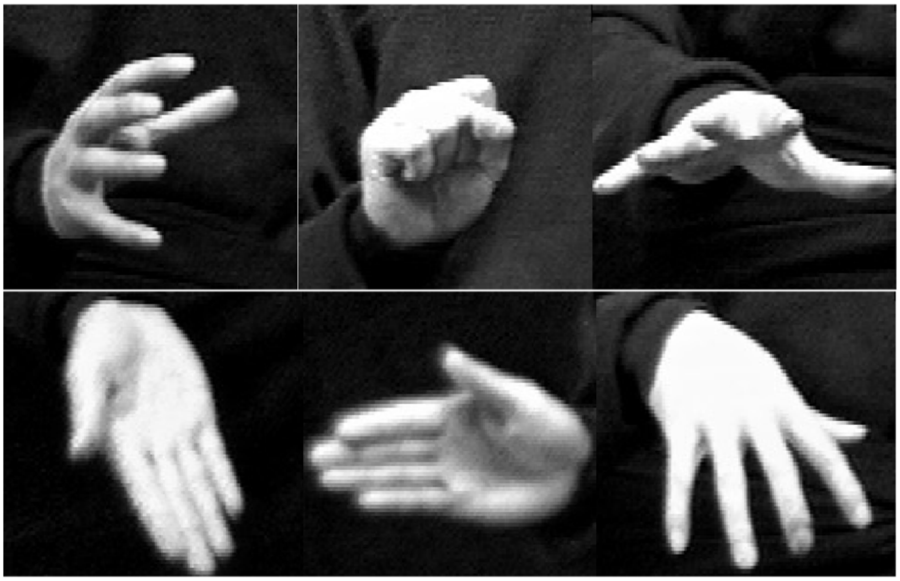

Fig. 1. Samples of hand configurations recognized in Ref. [16].

same size n. For each value of n, projections of the training set on the selected attribute subsets were created. OC1's *mktree* was invoked on each projection and the accuracy of the decision tree (measured as the percentage of correct classifications) induced by that projection was measured on the training set to be later used as weight in weighted voting. Then all 25 attribute subsets were used for voting on the test set. The random selection of a single attribute subset did *not* include replacement, i.e. there could be no repeated features in one subset, but the same features could be used in different voting subsets.

We performed both weighted and unweighted voting. In weighted voting, the weight of each attribute subset was set equal to its accuracy on the training set. To obtain statistically significant results, for each attribute subset of dimensionality n, the random selection and voting was repeated 30 times. This is, for any given n, 30 repetitions were done, each repetition consisting of randomly generating 25 subsets of size n and then voting with these subsets on both the training and the test sets. Figs. 2 and 3 present the results of this experiment on the training and the test set, respectively. To avoid undesirable bias, the weighted voting used the accuracies of the subsets on *training* data only, and these accuracies were used to vote on both the training and the test sets.

4.1.1. Discussion

Figs. 2 and 3 clearly show that weighted and unweighted voting give very similar accuracies. Weighted voting is slightly better for small attribute subset sizes, but in general the benefits of weighting for more than seven voting subsets are insignificant and, in a few cases, the results are poorer than those of unweighted voting. The dashed curve represents the accuracy achieved by the best decision tree built on one of the 25 random attribute subsets. It is clear that voting gives dramatic improvement in accuracy on the test set in comparison to a single best subset of attributes. The improvement is especially visible for small attribute subset sizes, and reaches as much as 7.5% for subsets of four attributes (Fig. 3).

Moreover, Figs. 2 and 3 demonstrate that the behavior of voting is qualitatively similar on the training and the test sets, with the difference between single best attribute subset accuracy and voting accuracy much more pronounced on the test set. Obviously, the accuracies achieved by all three methods are lower on the test set than on the training set.

As expected, the voting accuracy is small for small attribute subsets, but rises quite sharply as the attribute subset size increases. On the training set, the accuracy of weighted voting reaches a maximum for 14-attribute subsets (99.39%) and the accuracy of unweighted voting reaches maximum for 13 attribute subsets (99.43%). On the test set, the accuracy reaches a maximum for 13-attribute subsets in both weighted and unweighted voting (97.21% and 97.20%, respectively), and then declines for large attribute subsets, approaching the accuracy achieved by a *single run* of OC1 using *all* attributes (92.29%). This behavior is understandable: for small numbers of attributes the trees used for voting lack precision (they are too heavily "pruned"), and for large subsets the independence of all ensemble members decreases, resulting in poorer voting performance. From 8- to 18-attribute subsets, the voting accuracy is practically constant on the test set. Based on these results, we can conclude that for the hand-pose/configuration dataset the best

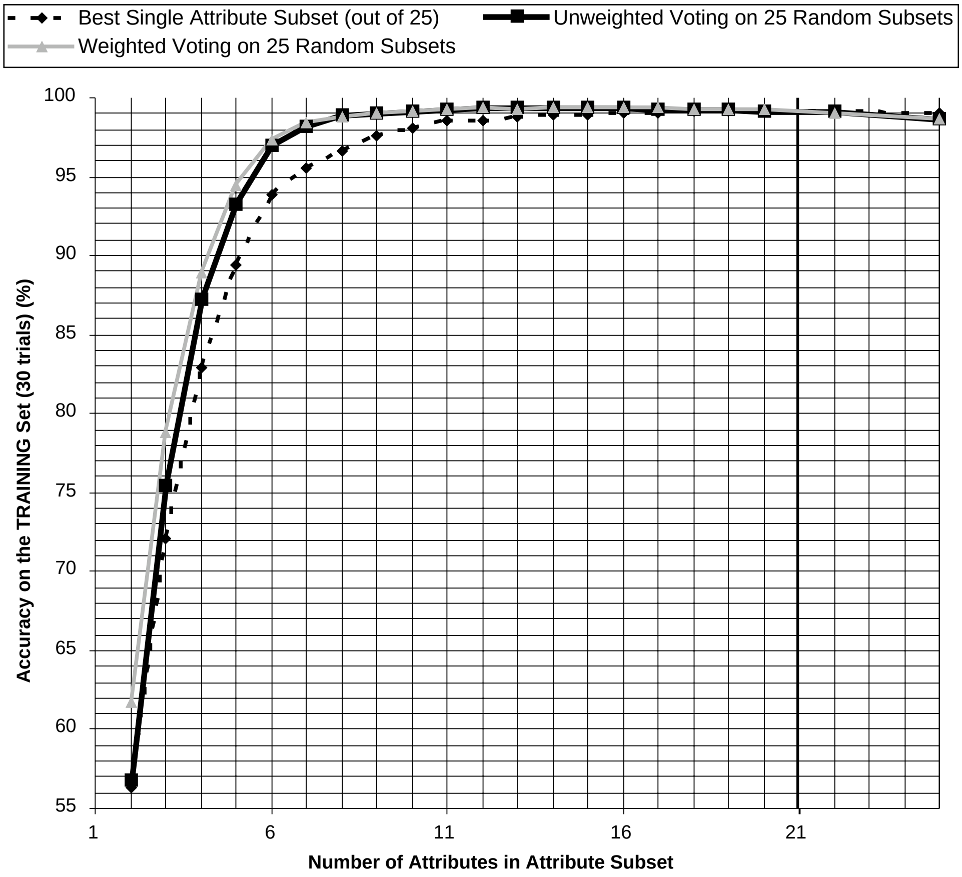

Fig. 2. Accuracy of AB (weighted and unweighted voting on 25 predictors) and accuracy of predictor induced by a single best subset of attributes as a function of the attribute subset size. Results computed for the training set.

voting accuracy of AB is achieved for attribute subset sizes between 1/3 and 1/2 of the total number of attributes. This is also intuitively correct, since this attribute subset dimensionality yields the voters that are diversified and independent, but at the same time not overly simplified or "pruned". The comparison of the best voting accuracy on the test set (97.21%) and the accuracy obtained by a single run of OC1 using all attributes in the database (92.29%) gives a 4.92% improvement, which is statistically significant with a high degree of confidence ($P < 0.001$).

4.1.2. Classification stability

When averaging accuracies of 30 trials for each attribute subset size, we also computed the standard deviation of these accuracies to obtain a measure of classification stability for various attribute subset sizes.[1] The results of the experiment, presented in Figs. 4 and 5, are also intuitively correct. The standard deviations of the accuracies behave in a very similar manner for weighted and unweighted voting in both training and test sets. They are almost equal for all tested attribute subset sizes, except for small subsets, where weighted voting seems to be more stable. The standard deviation for the best single attribute subset follows a curve similar in shape to those of weighted and unweighted voting. It decreases steadily, and would reach 0 for attribute subset sizes equal to the total number of attributes, since all the subsets would then induce identical predictors. The standard deviation of voting accuracy would also reach 0 for attribute subset sizes equal to the total number of attributes, since all voters would be the same and the results of all votes would be identical. The large discrepancy between the standard deviation of the single best attribute subset and that of unweighted/weighted voting for attribute subsets of size 1 can be explained as follows: we drew 25 attribute subsets randomly in each trial, and used them for voting. For all one-attribute subsets (out of 28 attributes), the likelihood of drawing the single attribute giving the best accuracy was high for each set of 25 selections. The algorithm kept finding the same (or "close") best single attribute subset, which yielded similar accuracy in the majority of the

[1] We used the unbiased estimates of standard deviation.

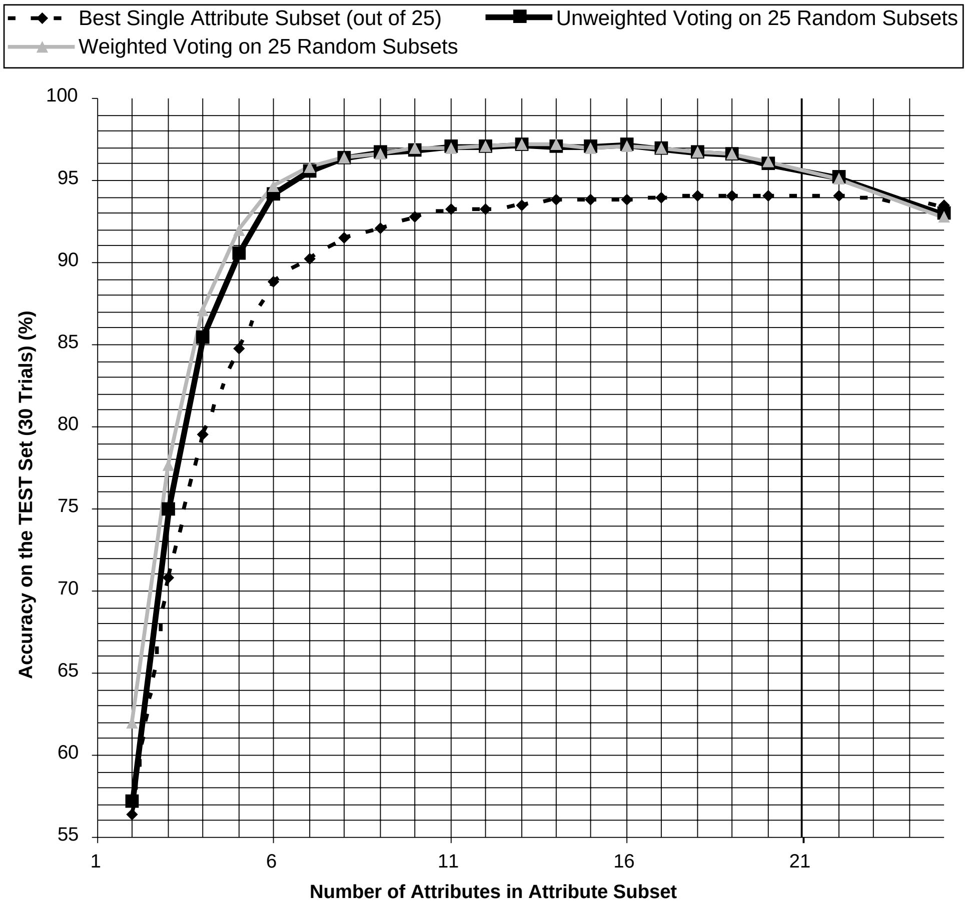

Fig. 3. Accuracy of AB (weighted and unweighted voting on 25 predictors) and accuracy of predictor induced by a single best subset of attributes as a function of the attribute subset size. Results computed for the test set.

30 trials. The standard deviation would most likely be equal to 0 if we drew more than 25 attribute subsets in each trial, since drawing the best single attribute out of 28 attributes would become almost certain for a large number of random selections. For two attribute subsets, the number of possible combinations is significantly larger and, therefore, the standard deviation of accuracy increases sharply from one- to two-attribute subsets.

4.1.3. Conclusion

From the results presented in Figs. 2–5, the range of attribute subset sizes giving the best combination of voting accuracy and voting stability lies between 8 and 18 attributes. This number is, obviously, likely to be tied to the particular dataset under study.

4.2. Increasing voting accuracy by ranking attribute subsets

As we stated in Section 1, it is possible to increase the ensemble's voting accuracy even further by selecting larger number of attribute subsets, ranking them by accuracy on training data or on a validation set and then using only the best subsets for voting on the test set. Table 2 shows the results of a series of experiments in which ranking was employed. In all experiments we drew 100 or 300 9-attribute subsets and selected the top 25 with highest accuracy on training data for unweighted voting. The results present the voting performance on the test set.

The key observation is that ranking the attribute subsets on the training set improves the accuracy of voting. The difference between the accuracy obtained for 25 best subsets out of 100 (97.40%) and for 25 subsets without ranking (96.74%), as in Section 4.1, is statistically significant ($P < 0.001$). Therefore, drawing more attribute subsets and using only the best for ensemble voting appears to improve the performance of the resulting classifier.

To further evaluate the subset ranking idea, we performed an additional experiment where we selected 300 attribute subsets and then picked the 25 best for voting. As shown in Table 2, the accuracy (97.51%) was better than for 100 random attribute subsets (97.40%), but

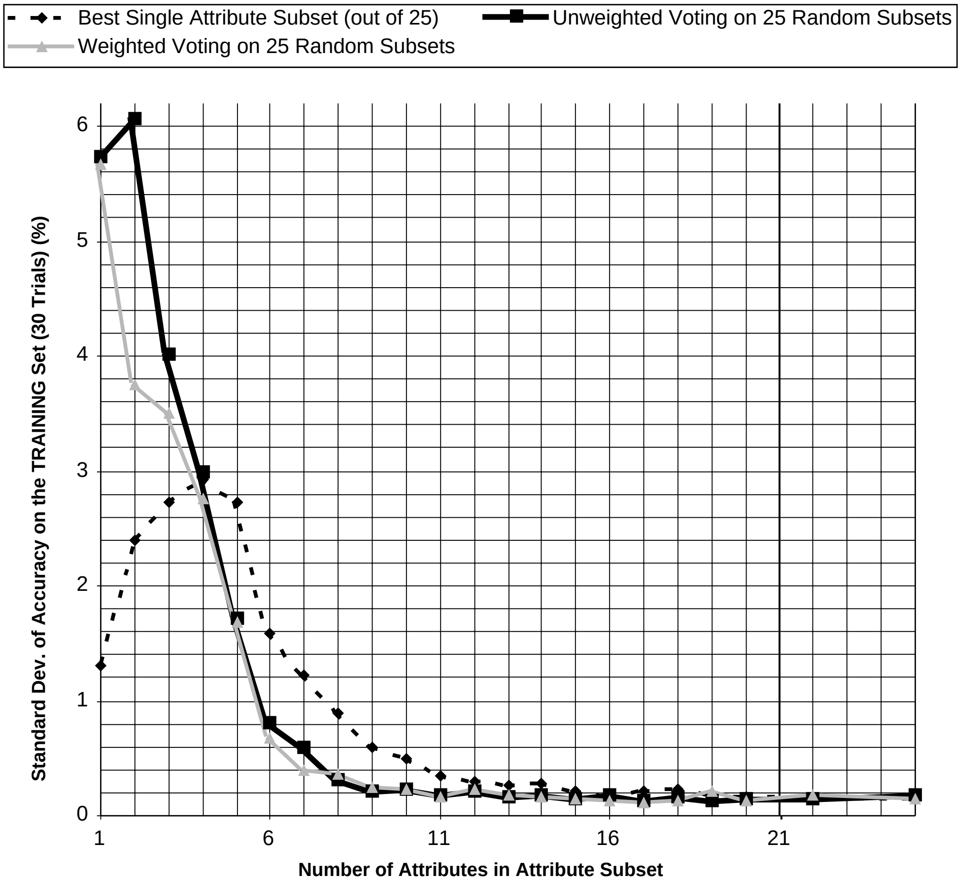

Fig. 4. Standard deviation of accuracy of AB (weighted and unweighted voting on 25 predictors) and standard deviation of accuracy of predictor induced by a single best subset of attributes as a function of the attribute subset size. Results computed for the training set.

the improvement had only moderate statistical significance ($0.01 < P < 0.05$). This suggests that the accuracy approaches some maximum value asymptotically. Interestingly, Breiman's bagging can also be augmented in a very similar manner by creating more samples of the training set than necessary and voting only on certain number of the best trees.

We must emphasize that ranking of attribute subsets is performed using only training data, and accuracy of the ensemble is measured only on test data. One may be tempted to establish the attribute ranking on test data, but the danger of this approach is that the weights would be obtained from the predictor accuracy on test data, and therefore the ranked/weighted vote would most likely overfit the test set and create a biased (optimistic) final accuracy estimate. To avoid this potential overfitting problem, the subsets should be ranked either on the training set (as we have done in these experiments) or on a validation set, separate from both the training and the test data.

4.3. *Attribute partitioning vs. data partitioning*

For the second set of experiments, we also use the fixed training and testing partitions of Ref. [16]. To limit the number of variables in the experiments, we fix the attribute subset size in AB at nine dimensions. This number lies inside the region of high accuracy and stability (see Figs. 2–5), is reasonably small, and is close to 1/3 of the total number of attributes, which we find appealing. For standard bagging, we implement the method described by Breiman and Quinlan in Refs. [3,4]. Both AB and standard bagging are compared for various numbers of voting decision trees. To compare the merits of "pure" attribute partitioning and data partitioning ensemble learning methods, we use unweighted voting both for AB and for bagging. Moreover, the number of voting trees in AB is always equal to the number of trees induced by randomly selected subsets of nine attributes, i.e. no "attribute subset ranking" is performed to boost the AB performance, as was done in Section 4.2. For each number of voting trees, the experiment was repeated 30 times both

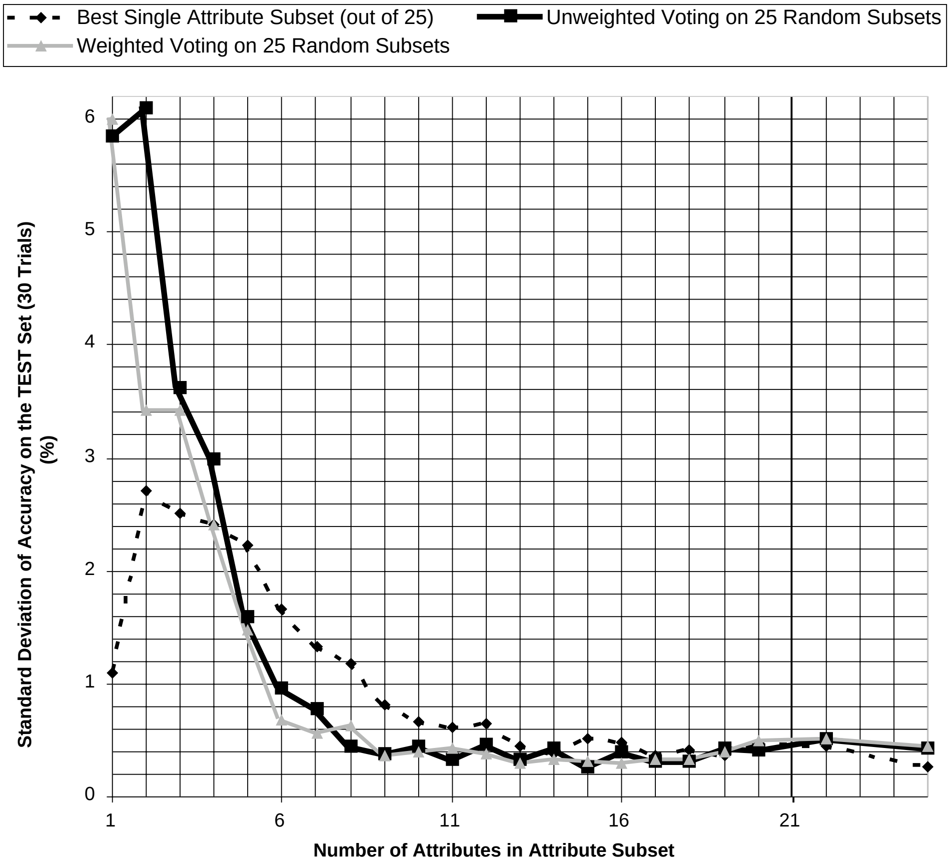

Fig. 5. Standard deviation of accuracy of AB (weighted and unweighted voting on 25 predictors) and standard deviation of accuracy of predictor induced by a single best subset of attributes as a function of the attribute subset size. Results computed for the test set.

Table 2
Results of unweighted voting on the test set by ranked attribute subsets

Subsets drawn	AB accuracy (%)	Std. dev. (%)
100	97.40	0.27
300	97.51	0.22

Twenty-five best 9-attribute subsets used for voting.

for AB and for bagging, and the average performance was computed. Fig. 6 shows the resulting accuracy for AB and bagging, whereas Fig. 7 shows the stability (standard deviation) of their voting accuracies.

4.3.1. *Discussion*

It can be easily seen in Fig. 6 that for small numbers of voting trees (3 and 5) Bagging performs better than AB. This seems intuitively correct because of the difference in the amount of data sampled from the training set by both methods: Breiman's bagging always samples the entire training set for each decision tree, whereas AB uses only about 1/3rd of the information in the training set (nine attributes out of 28) each time a decision tree is built. Hence, AB needs more voting trees in order to "cover" the dataset. However, starting from seven voters, AB starts to dominate, and for all numbers of voters greater than 9 it performs better than bagging with a high level of statistical significance ($P < 0.001$).[2] For both methods, the accuracy increases with the number of voters and seems to asymptotically converge to two *different* maximum values. The higher maximum value reached by AB supports the conclusion presented in Ref. [5] that attribute partitioning methods are superior to data partitioning methods.

Observation: We want to emphasize that the maximum accuracy achieved in this experiment by attribute bagging on 101 voters (97.19%) is actually lower than that achieved by the top 25 voters (97.40%), as described in Section 4.2. This difference is statistically significant ($P < 0.001$). The

[2] We ended our tests at 101 voting trees due to excessive computation times.

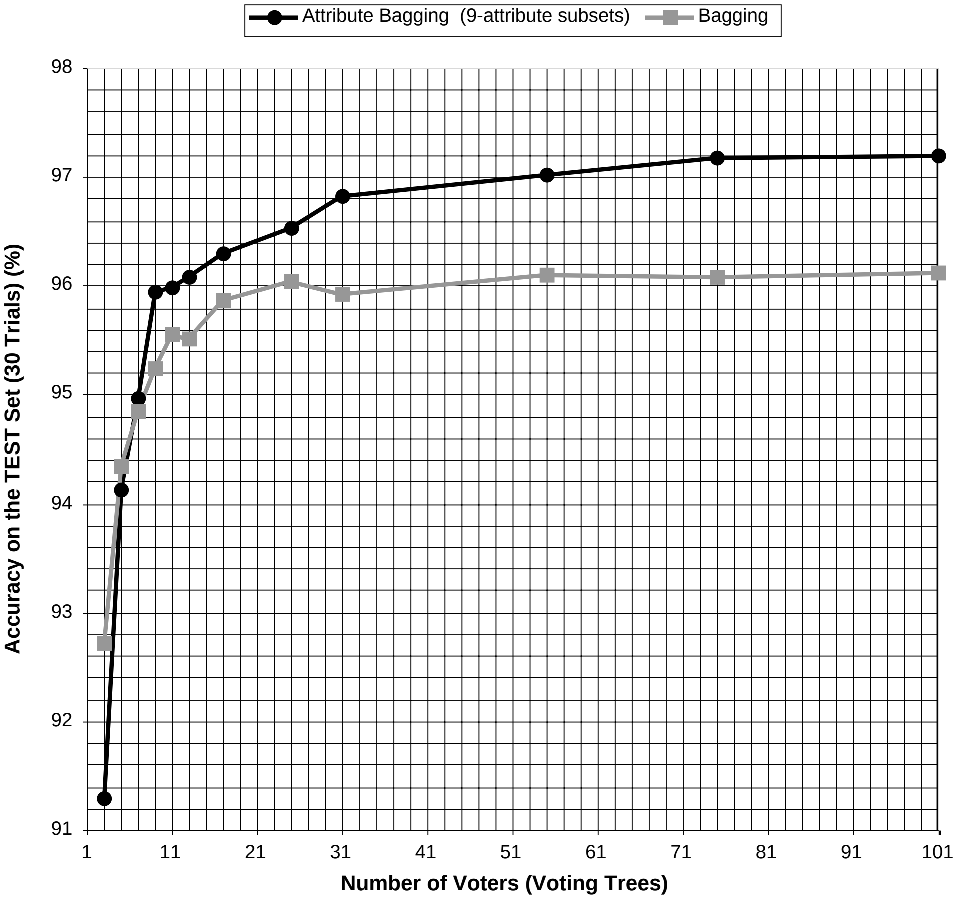

Fig. 6. Accuracies of AB and standard bagging as functions of the number of the voting trees.

training/test sets were the same, the sizes of voting attribute subsets were equal (9), and the only difference between the two cases was that in the experiments of Section 4.2 we drew 100 random attribute subsets and selected for voting only 25 with the best accuracies on the test set. Therefore, these results suggest that selecting *m* attribute subsets, ranking them by their accuracy on the training set, and then voting on *only* the *best k* subsets, where $k < m$, gives better ensemble accuracy than simply voting on classifiers induced by all *m* random attribute subsets.

In terms of voting stability, the plots in Fig. 7 also follow the intuition: the standard deviation drops as the number of voters increases. Both methods are roughly equivalent with respect to the standard deviation, although for small numbers of voting trees bagging gives lower standard deviations than AB. This phenomenon can probably be explained using the same argument as the one presented for voting accuracy above. Both standard deviation curves show a very similar valley for 25 voters and then an almost identical rise for 31 voters. Although we cannot fully explain this phenomenon, we hypothesize that it is related to the ratio between the number of voting trees and the number of classes in the dataset: for some values of this ratio, the stability of voting increases. The issue of finding an optimal number of voters for a given number of classes in ensemble learning deserves further investigation.

4.4. *AB vs. a single OC1 run on multiple holdouts*

In addition to testing AB on fixed training and test sets, we performed five holdout runs in which a certain percentage (50% or 70%) of the examples from the entire dataset were randomly selected for training and the rest used for testing. For each run (each training/test set pair), we computed the accuracy for AB and for a single run of OC1 on the full feature set. AB was run with 9-attribute subsets, 25 subsets selected initially and all 25 combined through unweighted voting. There was only one trial for each training/test set pair. The resulting accuracies are the average over five runs. The results are presented in Table 3. The first column of Table 3 shows the percentage of the dataset used for training. These results clearly show the superiority of ensemble

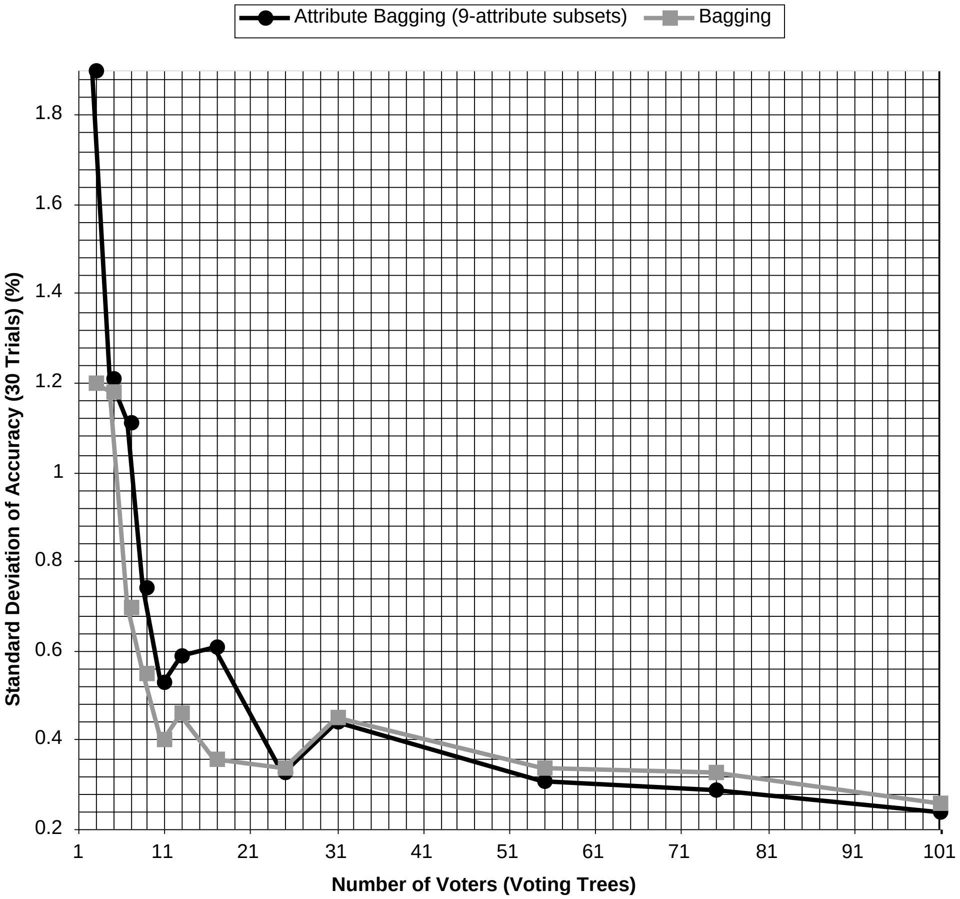

Fig. 7. Standard deviations of accuracies of AB and standard bagging as functions of the number of the voting trees.

Table 3
Averaged results of five holdout runs for single OC1 runs and AB

Training set (%)	OC1 acc. (%)	Std. dev. (%)	AB acc. (%)	Std. dev. (%)
70	92.18	1.00	96.17	0.30
50	91.12	0.89	95.77	0.76

Table 4
Averaged results of 10 holdout runs for bagging and AB

Bagging accuracy (%)	Std. dev. (%)	AB accuracy (%)	Std. dev. (%)
94.45	1.12	95.78	0.56

classifiers as compared to a single classifier. A t-test performed on the average accuracy of OC1 and AB indicates that the accuracy of AB is higher for both 70% and 50% training sets with 99.5% confidence. As expected, selecting 70% of the data as training set increases the accuracy on the test set for both OC1 and AB.

4.5. *AB vs. bagging on multiple holdouts*

Finally, we compared the performance of AB and bagging in 10 holdout runs where 50% of the entire dataset was selected randomly for training in each run. In each holdout both algorithms were run 30 times to obtain robust statistics. The voting was unweighted, and done for 55 trees in both algorithms. In both cases, only 55 different attribute projections/training sets were generated and *all* of them were used in voting. No attribute subset ranking was performed. The results are presented in Table 4. The average standard deviation computed for 30 repetitions in each holdout was 0.34% for AB and 0.40% for bagging. This indicates that both algorithms are similar in stability for *a given pair* of training/test sets, but bagging is more sensitive to the selection of test and training sets, as evidenced by its larger standard deviation in Table 4. In addition, AB is 1.33% better than Breiman’s bagging with 99.5% of confidence in a t-test.

Table 5
Comparison of best accuracies achieved by various algorithms on the hand-pose database

Algorithm	Best accuracy (%)
HCV	76.1
CN2	87.1
ID3	89.5
C4.5	90.1
NewID	91.0
RIEVL (Exact)	90.6
RIEVL (Flexible)	94.4
OC1 (on all attributes)	92.29
OC1 with bagging; 25 voters	96.05
OC1 with bagging; 101 voters	96.13
OC1 with AB; 25 9-attribute voters	**96.74**
OC1 with AB; 25 13-attribute voters	**97.21**
OC1 with AB; 101 9-attribute voters	**97.19**
OC1 with AB; 25 9-attr. voters out of 100	**97.40**
OC1 with AB; 25 9-attr. voters out of 300	**97.51**

Results for classifiers other than OC1 reprinted from Ref. [16].

5. Conclusions

We have described attribute bagging (AB), a technique for improving the accuracy and stability of classifier ensembles by voting on classifiers induced by (ranked) random attribute subsets. Before the classifier ensembles are built, our algorithm finds an appropriate attribute subset size by a random search in the feature subset dimensionality. The technique follows the "wrapper" model of feature subset selection and can be used with any classification algorithm. It can be applied to databases with large numbers of attributes.

AB was tested on a database of hand poses [16] and consistently performed better than bagging, HCV, CN2, ID3, C4.5, NewID, RIEVL, and OC1, as shown in Table 5. ID3, C4.5 and NewID are decision tree algorithms [18–20], whereas HCV, CN2 and RIEVL are rule induction learning systems [16,21,22].

Moreover, AB executed faster than bagging in our experiments, due to the smaller amount of data transferred and duplicated during creation of the training sets, as well as the faster induction of the decision trees. The training sets created by projections of the original training set on random subsets of attributes provide the induced predictors with full class representation of the original training set, whereas the training sets created by random selection of data points in bagging may underrepresent some classes, resulting in poorer predictors.

We also provided evidence that selecting m attribute subsets, ranking them by their accuracy on the training set, and then voting on *only* the *best* k subsets, where $k < m$, gives better ensemble accuracy than simply voting on classifiers induced by all m random attribute subsets.

6. Summary

We present attribute bagging (AB), a technique for improving the accuracy and stability of classifier ensembles induced using random subsets of features. AB is a wrapper method that can be used with any learning algorithm. It establishes an appropriate attribute subset size and then randomly selects subsets of features, creating projections of the training set on which the ensemble classifiers are built. The induced classifiers are then used for voting.

This article compares the performance of our AB method with bagging and single-classifier algorithms such as C4.5, RIEVL and OC1 on a hand-pose recognition dataset. It is shown that AB gives consistently better results than bagging, both in accuracy and stability. We present extensive tests and discussions of the accuracy and stability of ensemble voting in the AB method and in bagging for a wide range of attribute susbset sizes and varying number of voters, using both weighted and unweighted voting schemas. Also, our tests show that the two-stage approach, in which the appropriate attribute subset size is established first, and only the best (most accurate) random attribute subsets are used in voting, gives results better than those obtained with unranked random subsets. The paper supports the claim that attribute partitioning methods are superior to data partitioning methods in ensemble learning.

Acknowledgements

This research has been partially funded by the National Science Foundation KDI program, Grant No. BCS-9980054, "Cross-Modal Analysis of Signal and Sense: Multimedia Corpora and Tools for Gesture, Speech, and Gaze Research" and by NFS/CAREER Award No. 9984426.

The authors would also like to acknowledge the insightful comments of the anonymous reviewer, which helped improve the final version of this manuscript.

References

[1] R.O. Duda, P.H. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, New York, 2000.

[2] T.G. Dietterich, Machine learning research: four current directions, AI Magaz. 18 (1997) 97–136.

[3] L. Breiman, Bagging predictors, Mach. Learning 24 (1996) 123–140.

[4] J.R. Quinlan, Bagging, boosting, and C4.5, Proceedings of AAAI/IAAI (13th American Association for Artificial Intelligence National Conference on Artificial Intelligence, Portland, Oregon), Vol. 1, 1996, pp. 725–730.

[5] K. Tumer, J. Ghosh, Classifier combining: analytical results and implications, Working notes from the Workshop 'Integrating Multiple Learned Models', 13th National Conference on Artifical Intelligence, August 1996, Portland, Oregon.

[6] K. Tumer, N.C. Oza, Decimated input ensembles for improved generalization, Proceedings of the International Joint Conference on Neural Networks, Washington, DC, 1999.

[7] P. Langley, Selection of relevant features in machine learning, Proceedings of the AAAI Fall Symposium on relevance, New Orleans, LA, AAAI Press, 1994.

[8] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: W.W. Cohen, H. Hirsh (Eds.), Machine Learning: Proceedings of the 11th International Conference, Morgan Kaufmann Publishers, San Francisco, CA, 1994, pp. 121–129.

[9] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, Genetic Programming 1997: Proceedings of the 2nd Annual Conference, Morgan Kaufmann, Stanford University, CA, 1997, pp. 380–386.

[10] M.A. Hall, L.A. Smith, Feature subset selection: a correlation based filter approach, Proceedings of the Fourth International Conference on Neural Information Processing and Intelligent Information Systems (ICONIP'97), New Zealand, Springer, Berlin, 1997, Vol. 2, pp. 855–858.

[11] H. Liu, R. Setiono, A probabilistic approach to feature selection—a filter solution, Proceedings of the 13th International Conference on Machine Learning (ICML'96), Bari, Italy, 1996, pp. 319–327.

[12] Z. Zheng, G.I. Webb, Stochastic attribute selection committees, Proceedings of the Australian Joint Conference on Artificial Intelligence, Springer, Berlin, 1998, pp. 321–332.

[13] Z. Zheng, G.I. Webb, K.M. Ting, Integrating boosting and stochastic attribute selection committees for further improving the performance of decision tree learning, Proceedings of the 10th International Conference on Tools with Artificial Intelligence, Los Alamitos, CA, IEEE Computer Society Press, Silver Spring, MD, 1998, pp. 216–223.

[14] S.D. Bay, Combining nearest neighbor classifiers through multiple feature subsets, Proceedings of the 17th International Conference on Machine Learning, Madison, WI, 1998, pp. 37–45.

[15] E.B. Kong, T.G. Dietterich, Error-correcting output coding corrects bias and variance, Proceedings of the Twelfth National Conference on Artificial Intelligence, Morgan Kauffman, San Francisco, CA, 1995, pp. 313–321.

[16] Meide Zhao, F. Quek, Xindong Wu, RIEVL: recursive induction learning in hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1174–1185.

[17] S.K. Murthy, S. Kasif, S. Salzberg, A system for induction of oblique decision trees, J. Artif. Intell. Res. 2 (1994) 1–32.

[18] J.R. Quinlan, Induction of decision trees, Mach. Learning 1 (1986) 81–106.

[19] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, USA, 1993.

[20] R. Boswell, Manual for NewID Version 6.1, Technical Report TI/P2154/RAB/4/2.5, The Turing Institute, Glasgow, 1990.

[21] X. Wu, The HCV induction algorithm, in: S.C. Kwasny, J.F. Buck (Eds.), Proceedings of the 21st ACM Computer Science Conference, ACM Press, New York, 1993, pp. 168–175.

[22] P. Clark, T. Niblett, The CN2 induction algorithm, Mach. Learning 3 (1989) 261–283.

About the Author—ROBER BRYLL received the M.S. degree in Biomedical Engineering from the Warsaw University of Technology in 1993 and the M.S. degree in Computer Science from the University of Illinois at Chicago in 1998. He is currently a Ph.D. candidate in the Computer Science and Engineering Department at the Wright State University in Dayton, Ohio. He works as a Research Assistant in the Vision Interfaces and Systems Laboratory. His research interests include machine vision, vision-based object tracking, data mining, agent systems, gesture tracking and analysis, video database systems, automatic video segmentation, optical flow computation, user interface design, pattern recognition, database system design, and object-oriented design and programming.

About the Author—RICARDO GUTIERREZ-OSUNA received the B.S. degree in Industrial/Electronics Engineering from the Polytechnic University of Madrid in 1992, and the M.S. and Ph.D. degrees in Computer Engineering from North Carolina State University in 1995 and 1998, respectively, From 1998 to 2002 he served on the faculty at Wright State University. He is currently an assistant professor in the Department of Computer Science at Texas A&M University. His research interests include pattern recognition, machine learning, biological cybernetics, machine olfaction, speech-driven facial animation, computer vision, and mobile robotics.

About the Author—FRANCIS QUEK is currently an Associate Professor in the Department of Computer Science and Engineering at the Wright State University. He has formerly been affiliated with the University of Illinois at Chicago, the University of Michigan Artificial Intelligence Laboratory, the Environmental Research Institute of Michigan (ERIM) and Hewlett-Packard Human Input Division. Francis received both his B.S.E. summa caum laude (1984) and M.S.E. (1984) in electrical engineering from the University of Michigan in two years. He completed his Ph.D. C.S.E at the same university in 1990. He also has a Technician's Diploma Electronics and Communications Engineering from the Singapore Polytechnic (1978). Francis is a member of the IEEE and ACM. He is director of the Vision Interfaces and Systems Laboratory (VISLab) which he established for computer vision, medical imaging, vision-based interaction, and human–computer interaction research. He performs research in multimodal verbal/non-verbal interaction, vision-based interaction, multimedia databases, medical imaging, collaboration technology, and computer vision.