International Journal of Innovative Computing, Information and Control
Volume **7**, Number **11**, November **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 6247–6259

IGF-BAGGING: INFORMATION GAIN BASED FEATURE SELECTION FOR BAGGING

Gang Wang[1,2], Jian Ma[3] and ShanLin Yang[1,2]

[1]School of Management
[2]Ministry of Education Key Laboratory of Process Optimization and Intelligent Decision-Making
Hefei University of Technology
No. 193, Tunxi Road, Hefei 230009, P. R. China
wgedison@gmail.com; slyang@mail.hf.ah.cn

[3]Department of Information Systems
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong
isjian@cityu.edu.hk

Received July 2010; revised January 2011

Abstract. *Bagging is one of the older, simpler and better known ensemble methods. However, the bootstrap sampling strategy in bagging appears to lead to ensembles of low diversity and accuracy compared with other ensemble methods. In this paper, a new variant of bagging, named IGF-Bagging, is proposed. Firstly, this method obtains bootstrap instances. Then, it employs Information Gain (IG) based feature selection technique to identify and remove irrelevant or redundant features. Finally, base learners trained from the new sub data sets are combined via majority voting. Twelve datasets from the UCI Machine Learning Repository are selected to demonstrate the effectiveness and feasibility of the proposed method. Experimental results reveal that IGF-Bagging gets significant improvement of the classification accuracy compared with other six methods.*

Keywords: Ensemble learning, Bagging, Feature selection, Information gain

1. Introduction.

Research in the area of machine learning and data mining has achieved significant progress in the concept of learning from labeled instances. Although many efficient methods have been proposed, they have been limited to simple concepts or problems. Furthermore, numerous results suggest that learning more difficult concepts tends to be extremely difficult. Among the research directions, they have evolved to address these difficulties, which is ensemble learning [1,2]. A good understanding of how to build more sophisticated ensemble methods and exploit various possibilities of extracting information from the environment will move us to be closer to achieving the original intent of machine learning and data mining [2].

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning methods which try to learn one hypothesis from training data, ensemble learning tries to construct a set of hypotheses and combine them to use [3]. Learners composing an ensemble are usually called base learners. Ensemble methods have been approved theoretically and empirically to demonstrate the advantages over the individual base learner. Bagging [4] and boosting [5,6] are two popular ensemble methods to enforce weak base learners. The effectiveness of such methods comes primarily from the diversity caused by re-sampling the training set.

In practice, there are two basic requirements on the base learners for ensemble creation: diversity, i.e., the candidate base learners should be as diverse as possible, and accuracy, i.e., the base learners should more or less perform well [2,4]. However, to assess and control diversity of base learners and to find the trade-off between the accuracy and diversity is not a trivial task [1,2]. In bagging, for example, the only factor encouraging diversity between these base learners is the proportion of different instances in the training data sets. Although the base learners used in bagging are sensitive to small changes in data, the bootstrap sampling appears to lead to ensembles of low diversity compared with other ensemble methods. And lots of comparative studies have been done and can be found in [7-10]. It appears that, on average, AdaBoost, the most prominent member of boosting, is the best method although bagging and other ensemble methods have their application niches as well.

In order to enhance the performance of bagging, some studies have been investigated. However, these studies usually introduced diversity through different training data sets or different initial conditions. The use of different features has been relatively ignored. Recently, ensemble method using randomly selected subsets of features has been shown to improve the performance considerably [11]. Although a great deal of diversity is introduced, the accuracy is reduced. For this reason, the performance of these ensemble methods is rather unstable. In this study, we balance the degree of diversity and accuracy and propose a new ensemble construction method, called IGF-Bagging, which aims at building accurate and diverse base learners and enhancing the performance of bagging. The main heuristic consists in applying feature selection technique and reconstructing a feature set for each base learner. A general issue in machine learning and data mining is that using too many features in the learning task can be problematic, particularly if there are irrelevant or redundant features [12,13]. This can lead to over fitting, in which irrelevant or redundant features may exert undue influence on the classification decisions because of the finite size of training instances. In feature selection, there are two general strategies, namely the filter model and the wrapper model [14,15]. The former selects features by being guided by some significance measures, and the latter employs a learning algorithm to evaluate the selected feature subsets. As wrapper model has higher computational complexity than filter model and bagging has also large computational burden, we select filter model as feature selection technique in this study. In detailed procedures, IGF-Bagging obtains bootstrap instances firstly. Then, it employs feature selection technique to identify and remove irrelevant or redundant features. Finally, base learners trained from the new sub data sets are combined via majority voting. Twelve datasets from the UCI Machine Learning Repository are selected to demonstrate the effectiveness and feasibility of proposed methods. Empirical results show that IGF-Bagging is effective in building ensembles, whose performance is better than that of many other ensemble methods, e.g., bagging, boosting and random subspace.

The remainder of the paper is organized as follows. In Section 2, the background of feature selection is discussed firstly. Following above analyses, we propose a new ensemble construction approach, IGF-Bagging, based on the bagging and information gain based feature selection. In Section 3, we present the details of experimental design. Section 4 reports the experimental results. Based on the observations and results of these experiments, Section 5 draws conclusions and future research directions.

2. Information Gain Based Feature Selection for Bagging.

2.1. Feature selection.

Feature selection has been an active research area in machine learning and data mining communities [12]. The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. Feature selection reduces the dimensionality of feature space, removes redundant, irrelevant or noisy data. It brings the immediate effects for application: speeding up an algorithm, improving the data quality and thereof the performance of classifier [13].

Diverse feature selection techniques have been proposed in the machine learning and data mining literature. Feature selection techniques broadly falls into two categories based on their approach: wrapper model and filter model [14,15]. The wrapper model requires one predetermined learning algorithm in feature selection process. Features are selected based on their effect on the performance of learning algorithm. For each new subset of features, the wrapper model needs to train the classifier. It tends to find features better suited to the predetermined learning algorithm resulting in superior learning performance. However, the wrapper model is computationally more expensive [12,16]. The filter model relies on general characteristics of the training data to select a feature set without involvement of a learning algorithm. In the filter model, the feature set estimators evaluate features individually. A typical feature selection algorithm computes some relevance measure for each feature, mostly derived from statistical analysis of the data samples in the training data set, and then assigns it a score. Once the features are ranked, in the second phase of learning system, one is often interested in achieving maximum classification accuracy with minimum number of features [12,17].

Considering the computational complexity of bagging and the higher computational burden of wrapper model, Information Gain (IG) based feature selection, one of popular filter models, is adopted in our research [18,19]. The definition of IG is based on entropy. Entropy is a commonly used in the information theory measure, which characterizes the purity of an arbitrary collection of examples. It is in the foundation of the IG based feature selection. The entropy measure is considered as a measure of system's unpredictability. The entropy of Y is:

$$H(Y) = -\sum_{y\in Y} p(y)\log_2(p(y)) \tag{1}$$

where $p(y)$ is the marginal probability density function for the random variable Y. If the observed values of Y in the training data set S are partitioned according to the values of a second feature X, and the entropy of Y with respect to the partitions induced by X is less than the entropy of Y prior to partitioning, then there is a relationship between features Y and X. Then, the entropy of Y after observing X is:

$$H(Y/X) = -\sum_{x\in X} p(x)\sum_{y\in Y} p(y/x)\log_2(p(y/x)) \tag{2}$$

Given the entropy as a criterion of impurity in a training data set S, we can define a measure reflecting additional information about Y provide by X that represents the amount by which the entropy of Y decreases. This measure is known as IG. It is given by:

$$IG = H(Y) - H(Y/X) = H(X) - H(X/Y) \tag{3}$$

IG is a symmetrical measure (refer to Equation (3)). The information gained about Y after observing X is equal to the information gained about X after observing Y.

2.2. **IGF-Bagging.** The main objective of ensemble methods is to improve classification accuracy by aggregating the classifications of a diverse of base learners [1,2]. Previous researches have shown that an ensemble of base learners is often more accurate than any of the individual base learners in the ensemble. Two popular ensemble methods are bagging and boosting. They both employ re-sampling techniques to obtain different training data sets for each of base learners. Bagging (bootstrap aggregating) [4] generates different training data sets by randomly drawing replacements from the original data set. The base learners' outputs are then combined by using the majority voting with equal weight.

AdaBoost [5,6], short for Adaptive Boosting, is the most common implementation of the boosting algorithm. It performs several learning iterations based on the same training data set. The decisions of the base learners in the ensemble are combined by using weighted voting, where each weight depends on the error of the base learner on the training data set.

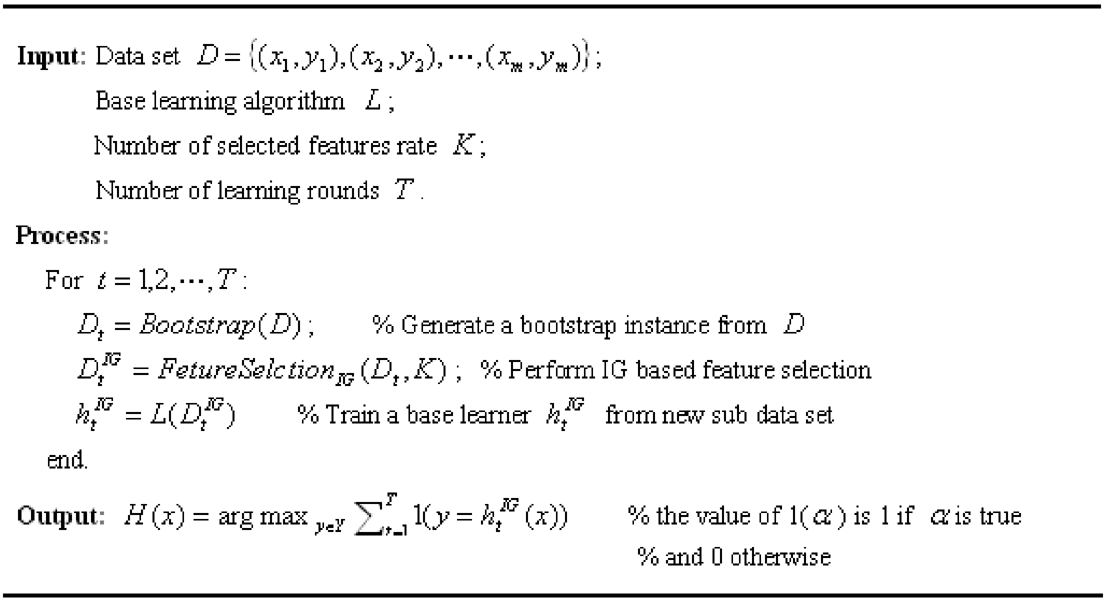
Recently, bias-variance decomposition of error has been used as a tool to study the behavior of ensemble methods and to develop new ensemble methods well suited to the bias-variance characteristics of base learners [20]. It has been shows that bagging can be expected to reduce the variance of a base learner. This is because bagging can be viewed as a method for developing a base learner that classifies using an estimate of the central tendency for the learner. In contrast, AdaBoost can reduce both bias and variance. And empirical studies showed that AdaBoost outperforms Bagging on average [7,10]. Although the bias-variance decomposition is able to explain the property of the ensemble learning, it is also a trick at present to construct an ensemble method. In practice, diversity and accuracy are two factors that should be taken care of while designing ensembles in order for them to generalize better. The success of AdaBoost has been explained, among others, with its diversity creating ability. Margineantu and Dietterich [21] devise the so-called "kappa-error" diagrams to show the effect of making the classifiers diverse at the expense of reduced individual accuracy. Plotting a diagram for an ensemble designed by Bagging and another designed by AdaBoost made the differences between the two approaches very clear. AdaBoost was creating inaccurate base learners by forcing them to concentrate on difficult objects and ignore the rest of the data. This, however, led to large diversity which boosted the ensemble performance, often beyond that of bagging.

Roughly speaking, ensemble methods can be divided into two classes: instance partitioning methods and feature partitioning methods. Bagging and boosting are all belongs to the former. Feature partitioning methods mainly include random subspace [11]. Random subspace perturbs the feature space to get diversity. In random subspace, a set of low dimensional subspaces is generated by randomly sampling from the original high dimensional feature vector and multiple classifiers constructed in random subspaces are combined in the final decision. In order to enforce the diversity of base learners in bagging, some explorations have been done from the perspective of integrating instance partitioning and feature partitioning [22,23]. Among them, a version of bagging called Random Forest was proposed by Breiman [22]. The ensemble consists of decision trees built again on bootstrap samples. The difference lies in the construction of the decision tree. The feature to split a node is selected as the best feature among a set of M randomly chosen features, where M is a parameter of the algorithm. This small alteration appeared to be a winning heuristic in that diversity was introduced without much compromising the accuracy of the individual base learners. Conversely, Random Forest is prone to over fitting in noisy classification tasks and it is ineffective when handling a large number of irrelevant features [24]. Next to this study, Rodriguez and Kuncheva proposed a rotation approach, named Rotation Forest [23], to encourage simultaneously individual accuracy and diversity within the ensemble. Diversity is promoted through the feature extraction for each base learner. To create the training data for base learners, the feature set is randomly split into K subsets and Principal Component Analysis is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Decision trees were also chosen as base learners because they are sensitive to rotation of the feature axes. Experimental results revealed that Rotation Forest construct individual base learners which are more accurate than these in AdaBoost and Random Forest, and more diverse than these in Bagging, sometimes more accurate as well. Like

Random Forest, Rotation Forest is also ineffective when encountering a large number of irrelevant features [24].

Through injecting feature partitioning strategy into bagging, base learners in bagging can get more diversity. The performance of ensemble methods is also enhanced. However, feature selection, anther form of feature partitioning, is seldom paid attention to. Feature selection has been a fertile field of research and development since 1970's in machine learning, pattern recognition and data mining [12]. Features that are irrelevant to learning tasks may deteriorate the performance of learning algorithms. Therefore, the omission of some features could not only be tolerable but even desirable relatively to the costs involved. And it can influence the quality of ensemble method in several ways, e.g., reducing learner complexity, promoting diversity of base learners, and affecting the trade-off between the accuracy and diversity of base learners [25]. In order to tackle the problem of irrelevant or redundant features, we introduce one of feature selection methods into bagging and propose a new version of bagging: IGF-Bagging.

IGF-Bagging combines one of the popular feature selection methods, IG based feature selection, with the standard bagging procedure. We want to utilize the feature selection, e.g., IG based feature selection, to enhance the accuracy and diversity of base learners. The proposed IGF-Bagging proceeds in a parallel of T rounds. In every round, a sub data set is bootstrap sampled with replacement firstly. Subsequently, the IG based feature selection method is employed to delete irrelevant or redundant features. Lastly, base learners trained with new sub data sets are combined by majority voting. The pseudo-code for the IGF-Bagging algorithm is given in Figure 1.



```
Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$;
       Base learning algorithm $L$;
       Number of selected features rate $K$;
       Number of learning rounds $T$.
Process:
  For $t = 1, 2, \cdots, T$:
    $D_t = Bootstrap(D)$;        % Generate a bootstrap instance from $D$
    $D_t^{IG} = FetureSelction_{IG}(D_t, K)$;  % Perform IG based feature selection
    $h_t^{IG} = L(D_t^{IG})$        % Train a base learner $h_t^{IG}$ from new sub data set
  end.
Output: $H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} 1(y = h_t^{IG}(x))$     % the value of $1(\alpha)$ is 1 if $\alpha$ is true
                                                              % and 0 otherwise
```

Figure 1. The IGF-Bagging algorithm

3. Experimental Design.

3.1. Data sets and evaluation criteria.

To demonstrate the effectiveness and feasibility of the proposed method, we selected twelve data sets from the University of California at Irvine (UCI) Machine Learning Repository [26]. A summary of these data sets is shown in Table 1.

The evaluation criteria of our experiments are adopted from the established standard measures in the fields of machine learning and data mining [4,5,11,22,23]. The definition

Table 1. Experimental data sets

Data set	Size	Attribute		Class
		Categorical	Continuous	
Audiology	226	0	69	24
Auto-mpg	398	2	5	4
Bridges2	108	10	1	6
Credit-a	653	6	9	2
Credit-g	100	7	13	2
Hayers-roth	132	4	0	3
Ionosphere	351	34	0	2
Machine	209	0	7	8
Mushroom	8124	0	22	2
Page-blocks	5473	0	10	5
Sonar	208	60	0	2
Splice	3177	0	60	3

of classification accuracy can be explained with respect to a confusion matrix as shown in Table 2.

Table 2. Confusion matrix

		Actual Condition	
		Positive	Negative
Test Result	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Formally speaking, classification accuracy is defined as follows:

$$Classification\ Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

3.2. **Experimental settings.** The experiments described in this section were performed on a PC with a 1.83 GHz Intel Core Duo CPU and 2GB RAM, using Windows XP operating system. Data mining toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.6.0 is used for experiment. WEKA is an open source toolkit, and it consists of a collection of machine learning algorithms for solving data mining problems [31].

In this study, two popular methods are chosen as the base learners, i.e., Decision Tree (DT) and Naïve Bayes (NB). Consequently, the evaluated methods are divided into two groups. The first group includes the standard DT, DT (Filtered), Bagging DT, Bagging (Filtered) DT, Boosting DT, Random Subspace DT and IGF-Bagging DT. The second group includes the standard NB, NB (Filtered), Bagging NB, Bagging (Filtered) NB, Boosting NB, Random Subspace NB and IGF-Bagging NB. The DT (Filtered) and NB (Filtered) mean that DT and NB are trained by filtered data sets with IG based feature selection. The Bagging (Filtered) DT and Bagging (Filtered) NB mean that Bagging DT and Bagging NB are also trained by filtered data sets with IG based feature selection.

For implementation of DT and NB, we chose J48 (WEKA's own version of C4.5) module and NaiveBayes module. And for implementation of ensemble learning, i.e., Bagging, Boosting and Random Subspace, we chose Bagging module, ADBoostM1 module and RandomSubSpace module. Besides above modules, the other modules were implemented in Eclipse using WEKA Package, i.e., WEKA.JAR. Following previous research

[4,5,11,22,23], ensembles of size 50 were used as a compromise between greater compute times required by larger ensembles and the ever-decreasing average-case marginal improvement in error that can be excepted from larger ensemble sizes. The feature selection ratio $K = 0.6$. Except when stated otherwise, all the default parameters in WEKA were used.

To minimize the influence of the variability of the training set, ten times 10-fold cross validation is performed on the twelve data sets. In detail, each dataset is partitioned into ten subsets with similar sizes and distributions. Then, the union of nine subsets is used as the training set while the remaining subset is used as the test set, which is repeated for ten times such that every subset has been used as the test set once. The average test result is regarded as the result of the 10-fold cross validation. The whole above process is repeated for 10 times with random partitions of the ten subsets, and the average results of these different partitions are recorded.

Following [27], we carry out as a first step an Iman-Davenport test [28], to ascertain whether there are significant differences among all the methods. Then, pairwise differences are measured using a Wilcoxon test. This test is recommended because it was found to be the best one for comparing pairs of algorithms [27]. The formulation of the test [29] is the following. Let d_i be the difference between the error values of the methods in ith data set. These differences are ranked according to their absolute values; in case of ties, an average rank is assigned. Let R^+ be the sum of ranks for the data sets on which the second algorithm outperformed the first, and let R^- be the sum of ranks where the first algorithm outperformed the second. Ranks of are split evenly among the sums

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{5}$$

and

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i). \tag{6}$$

Let T be the smaller of the two sums and N be the number of data sets. For a small N, there are tables with the exact critical values for T. For a larger N, the statistics

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{7}$$

is distributed approximately according to $N(0,1)$. We combine these two tests to assess the differences in performance of the different algorithms. When the comparison is between two algorithms only the Wilcoxon test is used.

Besides the Wilcoxon test, we also employ the statistics used in [30] to compare two learning algorithms across all data sets, namely the win/draw/loss record. The win/draw/loss record presents three values, the number of data sets for which algorithm A obtained better, equal, or worse performance than algorithm B with respect to classification accuracy. We also report the statistically significant win/draw/loss record; where a win or loss is only counted if the difference in values is determined to be significant at the 0.05 level by a paired t-test.

4. **Experimental Results.**

Our goal in this empirical evaluation is to show that IGF-Bagging is a plausible method. Stronger statements can only be made after a more extensive empirical evaluation. Tables 3 and 4 present classification accuracy of base learners and that of the compared methods, where the values following "±" are standard deviations. Generally speaking, the results obtained from the two tables show that the performance of the proposed IGF-Bagging method is better than the performance of the other methods.

Subsequently, several findings can be observed from Tables 3 and 4. Firstly, we consider the results of DT as base learner. As shown in Table 3, IGF-Bagging DT has the highest classification accuracy of 85.76%. Closely following IGF-Bagging DT is Boosting DT with a classification accuracy of 85.22%, Random Subspace DT with 85.13%, respectively. As we have expected, DT gets the lowest classification accuracy of 82.68%. Next, turning our attention to another base learner, NB, IGF-Bagging NB also gets highest classification accuracy of 80.93%. Closely following IGF-Bagging NB is also Boosting NB with a classification accuracy of 79.91%.

Table 3. Classification accuracy of different methods (DT as base learner)

Data set	DT(%)	DT (Filtered)	Bagging	Bagging (Filtered)	Boosting	Random Subspace	IGF-Bagging
Audiology	77.34±7.48	78.20±6.79	81.63±7.12	80.83±7.15	84.59±7.30	81.17±6.46	81.41±7.00
Auto-mpg	79.37±5.31	85.35±5.31	81.99±5.92	85.75±5.03	80.23±5.65	83.42±5.60	87.81±4.93
Bridges2	65.94±11.05	67.09±11.90	69.86±11.02	69.48±12.11	67.14±12.44	68.77±10.83	70.48±11.88
Credit-a	85.68±4.03	85.35±4.71	86.58±3.73	86.03±4.11	86.12±3.75	86.20±3.78	86.19±3.96
Credit-g	71.21±3.23	72.29±3.63	74.83±3.32	74.57±3.01	74.45±3.57	75.54±3.05	74.84±3.75
Hayers-roth	69.42±10.46	66.54±10.60	70.56±11.65	70.99±10.87	69.52±10.98	68.55±11.71	72.08±10.36
Ionosphere	89.88±4.34	91.54±4.22	92.28±4.43	93.10±4.24	94.10±4.07	93.53±3.94	93.16±4.05
Machine	88.90±5.88	86.28±8.50	89.72±6.05	87.66±7.22	90.37±5.27	90.29±5.47	90.06±5.96
Mushroom	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Page-blocks	96.91±0.68	96.93±0.67	97.36±0.62	97.20±0.65	97.00±0.67	97.38±0.57	97.43±0.56
Sonar	73.39±8.58	74.80±8.92	79.79±8.51	80.88±8.49	85.02±6.39	81.10±7.62	81.06±9.36
Splice	94.09±1.28	94.20±1.25	94.52±1.28	94.66±1.21	94.06±1.34	95.54±1.18	94.57±1.30
Average	82.68±12.63	83.21±12.79	84.93±11.65	85.10±11.65	85.22±11.81	85.13±12.00	85.76±11.91

Tables 3 and 4 verify the effectiveness of the proposed IGF-Bagging method. Based on the classification accuracy, we can judge which method is the best and which method is the worst. However, it is unclear what the differences between good and bad method are. And in order to ensure that the assessment does not happen by chance, we conducted statistical test to examine whether the proposed IGF-Bagging significantly outperforms the other six methods listed in this paper. Tables 5 and 6 show the comparison among the methods. Rows labeled s present the win/draw/loss record, where the first value is the number of the data sets for which $row < col$, the second is the number for which $row = col$, and the last is number for which $row > col$. Rows labeled p_w present the results of Wilcoxon test. For all the methods, the Iman-Daveport test has a p-value of 0.000, showing significant differences among them. As shown in Tables 5 and 6, our proposed IGF-Bagging is significantly better than other six methods.

In the experiments reports previously, we have chosen a value of feature selection ratio $K = 0.6$. A sensible question might be: which is the optimal value of K? However, there is not a value of K that we can consider optimal. Thus, the impact of using different K values is studied further. We varied K from 0.5 to 0.9 with interval 0.1. Figures 2 and

Table 4. Classification accuracy of different methods (NB as base learner)

Data set	NB	NB (Filtered)	Bagging	Bagging (Filtered)	Boosting	Random Subspace	IGF-Bagging
Audiology	72.86±6.37	74.32±6.46	72.19±6.55	74.18±6.28	73.42±7.83	70.70±6.66	74.72±6.44
Auto-mpg	67.49±6.36	65.22±6.02	67.21±6.32	65.17±6.10	67.49±6.36	66.53±6.36	68.12±5.94
Bridges2	68.01±11.61	66.16±12.38	67.91±11.70	65.86±11.87	67.86±12.78	66.91±10.84	68.28±11.56
Credit-a	77.84±4.05	77.55±3.59	78.10±4.15	77.91±3.84	81.46±3.88	77.42±3.96	77.99±3.77
Credit-g	74.90±3.73	74.44±3.60	74.88±3.74	74.51±3.57	74.89±3.97	74.29±3.49	75.87±3.80
Hayers-roth	81.78±8.11	82.62±8.01	81.51±8.36	82.77±8.41	80.55±9.95	80.27±10.69	83.77±8.41
Ionosphere	82.53±7.02	84.13±6.97	82.30±7.01	84.18±6.87	84.40±4.66	83.13±6.67	84.84±6.77
Machine	81.14±8.78	82.21±8.38	80.37±9.27	82.44±9.36	81.14±8.78	82.90±8.37	83.88±8.08
Mushroom	95.76±0.71	95.90±0.75	95.75±0.71	95.89±0.74	96.00±0.00	96.20±1.44	96.89±0.74
Page-blocks	89.92±2.48	91.28±1.80	90.08±2.16	91.41±1.64	90.18±1.84	92.16±1.69	92.19±1.51
Sonar	68.00±10.06	65.66±11.95	68.87±10.39	66.28±11.74	68.35±8.11	68.49±9.71	68.52±11.21
Splice	95.44±1.10	95.93±1.07	95.44±1.08	95.94±1.07	93.23±1.33	95.33±1.07	96.14±1.04
Average	79.64±11.77	79.62±12.68	79.55±11.85	79.71±12.69	79.91±11.64	79.53±12.34	80.93±12.05

3 display the classification accuracy curve for seven methods when K varies from 0.5 to 0.9. As shown in Figures 2 and 3, we can see clearly that IGF-Bagging gets the highest values of classification accuracy among seven methods. In addition, IGF-Bagging is less sensitive to K than DT (Filtered), Bagging (Filtered) and Random Subspace, especially for the DT as base learner. These results further prove that IGF-Bagging, integrating feature selection with bagging, can enhance the performance of base learners.

According to the above experimental results, we can draw the following conclusions:

(1) The integration of feature selection and bagging has brought significant performance improvements as shown in Tables 3-6 and Figures 2 and 3. The reason may be that IFG-Bagging outperforms bagging in generating more diverse and accurate base learners while outperforming random subspace in generating more accurate base learners. Features that are irrelevant or redundant to learning tasks may deteriorate the performance of learning algorithms. Therefore, feature selection can influence the quality of ensemble methods.

(2) From the ensemble learning perspective, in this research, we compare IGF-Bagging with other three ensemble methods, i.e., bagging, boosting and random subspace. Our empirical results are in conformity with previous studies [7-10]. It appears that, on average, boosting is the best method although bagging and random subspace have their application niches as well.

(3) From the base learner perspective, in this study, we use two different base learners, i.e., DT and NB. DT related ensemble methods get more impressive results than NB related ensemble methods. The main reason is that bagging and boosting is not so effective on stable learners [4,5]. However, NB is a stable learner [32] and the performance of NB is not able to be improved by ensemble methods as much as the other learners, e.g., DT.

(4) Feature selection is a hot topic in machine learning and data mining. Our empirical results show that feature selection could enhance the performance of classifiers. For example, the classification accuracy of DT is enhanced from 82.68% to 83.21% after using IG based feature selection. For bagging, the classification accuracy is also enhanced from

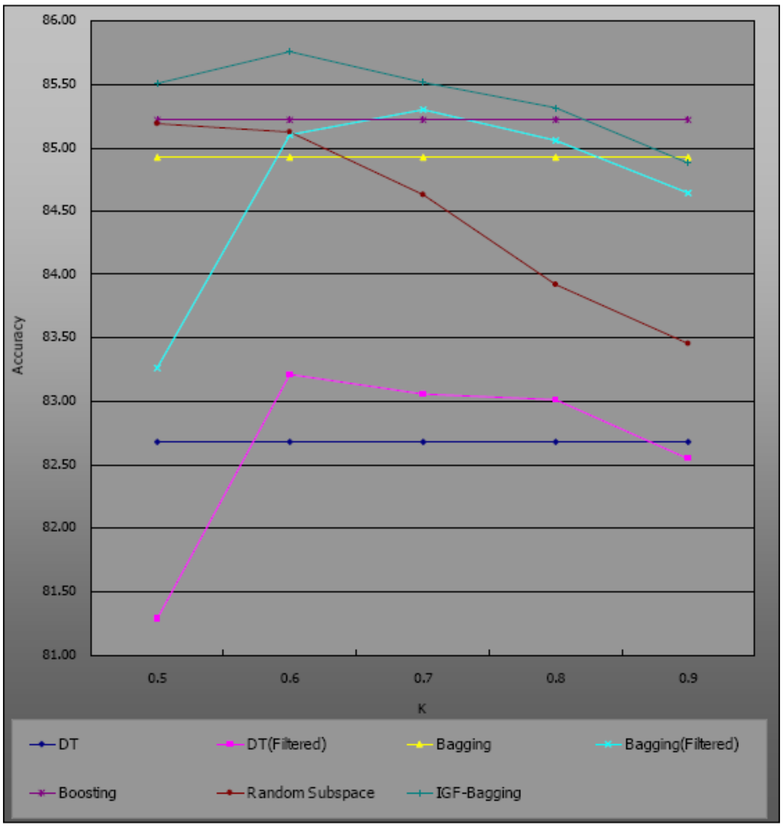


Figure 2. Classification accuracy at different K (DT as base learner)

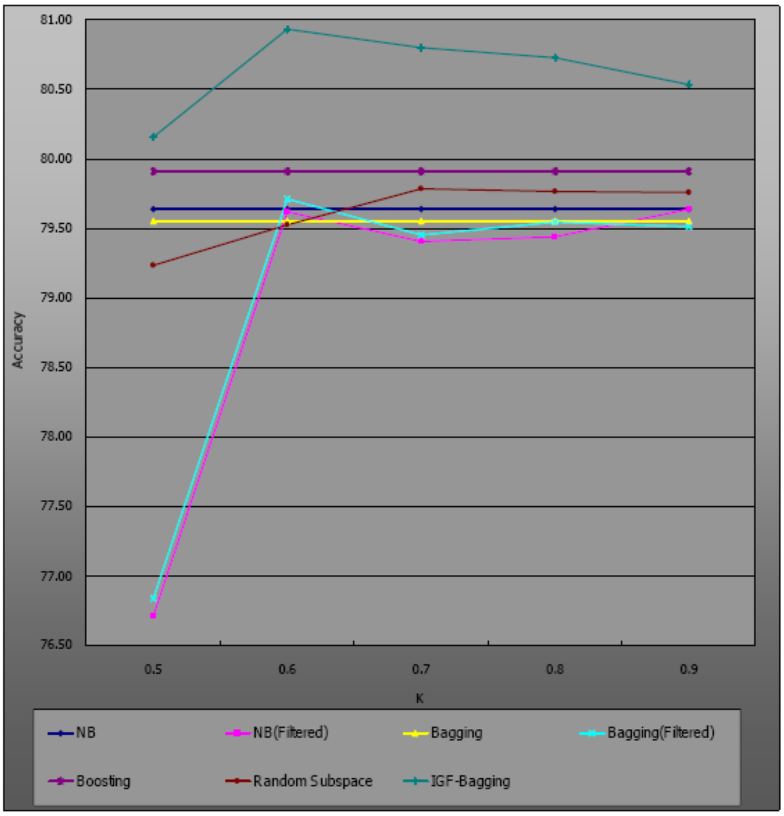


Figure 3. Classification accuracy at different K (NB as base learner)

Table 5. Significant test results (DT as base learner)

Method	DT	DT (Filtered)	Bagging	Bagging (Filtered)	Boosting	Random Subspace	IGF-Bagging
Mean All	82.68	83.21	84.93	85.10	85.22	85.13	85.76
DT	s	7/2/3	11/1/0	10/1/1	9/2/1	10/1/1	11/1/0
	p_w	2.788**	13.944**	9.784**	12.734**	10.781**	12.000**
DT(Filtered)	s		10/1/1	10/1/1	8/2/2	10/1/1	11/1/0
	p_w		6.256**	7.000**	7.437**	8.290**	9.093**
Bagging	s			4/3/5	4/1/7	6/2/4	7/4/1
	p_w			1.089	1.699*	1.106	3.194**
Bagging(Filtered)	s				4/3/5	6/3/3	7/4/1
	p_w				0.277	0.768	2.535**
Boosting	s					5/3/4	6/3/3
	p_w					-0.246	2.270*
Random Subspace	s						5/4/3
	p_w						3.416**

Notes: [1] *P-values significant at alpha=0.05; **P-values significant at alpha=0.01. [2] Iman-Davenport test: 0.000

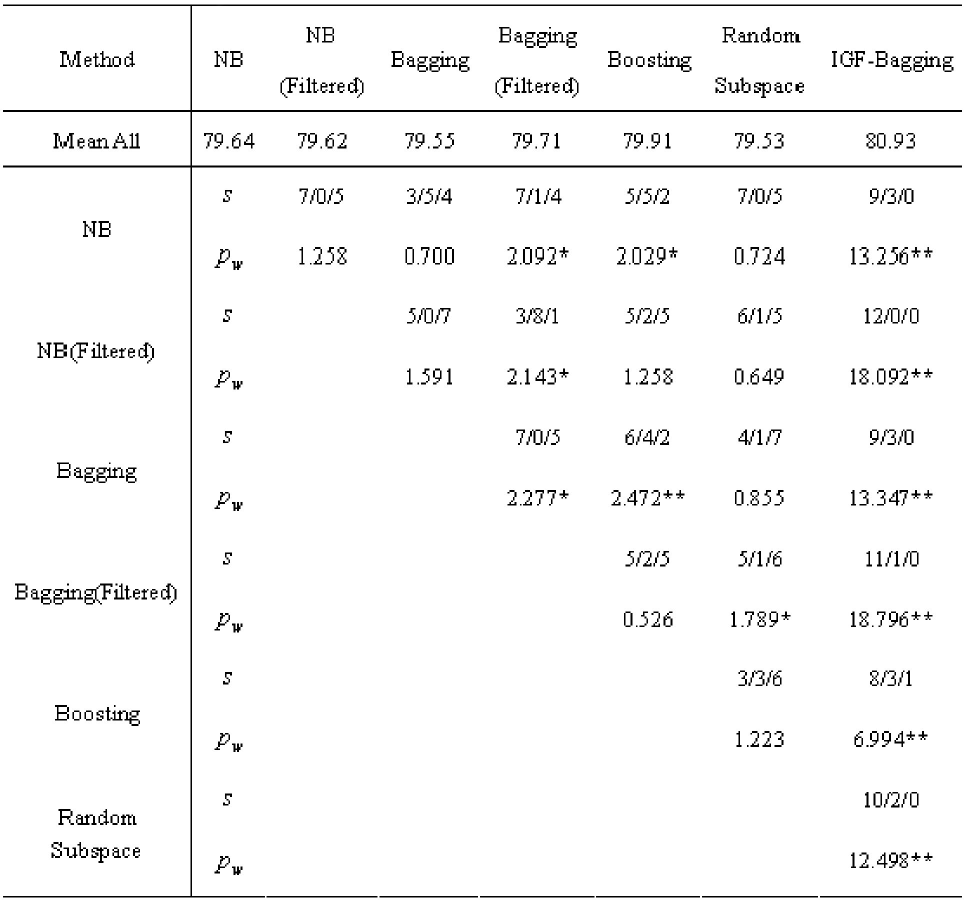
84.93% to 85.10% and from 79.55% to79.71% when using DT and NB as base learners, respectively.

5. Conclusions and Future Directions.

Ensemble learning trains multiple base learners and then combines their outputs. Since the generalization ability of ensemble learning could be significantly better than that of a base learner, ensemble learning has been a hot topic during the past years. In this paper, a new ensemble construction method, IGF-Bagging, is proposed based on bagging and IG based feature selection. After bootstrap sampling, this method applies IG based feature selection technique to each sub data sets, then identifies and removes irrelevant or redundant features, and finally combines base learners trained by new sub data sets via majority voting. Twelve datasets from the UCI Machine Learning Repository are selected to demonstrate the effectiveness and feasibility of proposed methods. Empirical results show that although this improvement for bagging is simple, it can effectively improve the classification accuracy.

Several future research directions also emerge. Firstly, large data sets for experiments should be collected to further valid the conclusions of this study. Secondly, further analyses are encouraged to explore the reasons why the integration of feature selection and bagging gets the better results, e.g., bias-variance analysis. Thirdly, since the bagging algorithm can be implemented in parallel in a straightforward manner, in the future, research parallel computing technique can be introduced to tackle the computational burden

Table 6. Significant test results (NB as base learner)



Method	NB	NB (Filtered)	Bagging	Bagging (Filtered)	Boosting	Random Subspace	IGF-Bagging
Mean All	79.64	79.62	79.55	79.71	79.91	79.53	80.93
NB	s	7/0/5	3/5/4	7/1/4	5/5/2	7/0/5	9/3/0
	p_w	1.258	0.700	2.092*	2.029*	0.724	13.256**
NB(Filtered)	s		5/0/7	3/8/1	5/2/5	6/1/5	12/0/0
	p_w		1.591	2.143*	1.258	0.649	18.092**
Bagging	s			7/0/5	6/4/2	4/1/7	9/3/0
	p_w			2.277*	2.472**	0.855	13.347**
Bagging(Filtered)	s				5/2/5	5/1/6	11/1/0
	p_w				0.526	1.789*	18.796**
Boosting	s					3/3/6	8/3/1
	p_w					1.223	6.994**
Random Subspace	s						10/2/0
	p_w						12.498**

Notes: [1] *P-values significant at alpha=0.05; **P-values significant at alpha=0.01. [2] Iman-Davenport test: 0.000

problem of bagging and feature selection. Fourthly, the role of feature selection in IGF-Bagging is in fact a specific scheme for perturbing the input features to introduce more diversity and accuracy. Exploring other feature selection technique and other efficient and effective schemes for perturbing the input features for bagging is anther interesting problem for future work.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (No. 71071045), the National High-tech R&D Program of China (863 Program) (No. 2009AA043403) and the Doctoral Special Fund of Hefei University of Technology (No. 2010HGBZ0607). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] R. Polikar, Ensemble based systems in decision making, *IEEE Circuits and Systems Magazine*, vol.6, pp.21-45, 2006.

[2] T. G. Dietterich, Machine learning research: Four current directions, *AI Magazine*, vol.18, pp.97-136, 1997.

[3] Z. H. Zhou, *Encyclopedia of Database Systems*, Springer, Berlin, 2009.

[4] L. Breiman, Bagging predictors, *Machine Learning*, vol.24, pp.123-140, 1996.

[5] Y. Freund and R. Schapire, Experiments with a new boosting algorithm, *Proc. of the 13th International Conference on Machine Learning*, Bari, Italy, pp.148-156, 1996.

[6] R. E. Schapire, The strength of weak learnability, *Machine Learning*, vol.5, pp.197-227, 1990.

[7] R. E. Banfield, L. O. Hall, K. W. Bowyer, D. Bhadoria, W. P. Kegelmeyer and S. Eschrich, A comparison of ensemble creation techniques, *Proc. of the 5th International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp.223-232, 2004.

[8] E. Bauer and R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, *Machine Learning*, vol.36, pp.105-139, 1999.

[9] T. G. Dietterich, Ensemble methods in machine learning, *Proc. of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp.1-15, 2000.

[10] T. K. Ho, A data complexity analysis of comparative advantages of decision forest constructors, *Pattern Analysis and Applications*, vol.5, pp.102-112, 2002.

[11] T. K. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, pp.832-844, 1998.

[12] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.

[13] A. L. Blum and P. Langley, Selection of relevant features and examples in machine learning, *Artificial Intelligent*, vol.97, pp.245-271, 1997.

[14] A. Jain and D. Zongker, Feature selection: Evaluation application and small sample performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, pp.153-158, 1997.

[15] R. Kohavi and G. John, Wrapper for feature subset selection, *Artificial Intelligence*, vol.97, pp.273-324, 1997.

[16] S. Hengpraprohm and P. Chongstitvatana, Feature selection by weighted-SNR for cancer microarray data classification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4627-4636, 2009.

[17] W. Y. Liu, L. Chi and B. W. Wang, Improved CPD feature selection research in text categorization, *ICIC Express Letters*, vol.3, no.4(B), pp.1423-1428, 2009.

[18] M. Dash and H. Liu, Feature selection for classification, *Intelligent Data Analysis*, vol.1, pp.131-156, 1997.

[19] G. Forman, An extensive empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research*, vol.3, pp.1289-1305, 2003.

[20] R. Kohavi and D. Wolpert, Bias plus variance decomposition for zero-one loss functions, *Proc. of the 13th International Conference on Machine Learning*, Bari, Italy, pp.275-283, 1996.

[21] D. D. Margineantu and T. G. Dietterich, Pruning adaptive boosting, *Proc. of the 14th International Conference on Machine Learning*, Nashville, TN, pp.211-218, 1997.

[22] L. Breiman, Random forests, *Machine Learning*, vol.45, pp.5-32, 2001.

[23] J. Rodriguez, L. Kuncheva and C. Alonso, Rotation forest: A new classifier ensemble method, *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol.28, pp.1619-1630, 2006.

[24] M. Gashler, C. Giraud-Carrier and T. Martinez, Decision tree ensemble: Small heterogeneous is better than large homogeneous, *Proc. of the 7th International Conference on Machine Learning and Applications*, San Diego, CA, pp.900-905, 2008.

[25] M. Bacauskiene, A. Verikas, A. Gelzinis and D. Valincius, A feature selection technique for generation of classification committees and its application to categorization of laryngeal images, *Pattern Recognition*, vol.42, pp.645-654, 2009.

[26] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[27] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, vol.7, pp.1-30, 2006.

[28] R. L. Iman and J. M. Davenport, Approximations of the critical regions of the Friedman statistic, *Communications in Statistics*, vol.6, pp.571-595, 1980.

[29] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics*, vol.1, pp.80-83, 1945.

[30] G. I. Webb, MultiBoosting: A technique for combining boosting and wagging, *Machine Learning*, vol.40, pp.159-196, 2000.

[31] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Boston, 2005.

[32] A. Tsymbal, S. Puuronen and D. W. Patterson, Ensemble feature selection with simple Bayesian classification, *Information Fusion*, vol.4, pp.87-100, 2003.