Bootstrap Feature Selection for Ensemble Classifiers

Rakkrit Duangsoithong and Terry Windeatt

Center for Vision, Speech and Signal Processing
University of Surrey
Guildford, United Kingdom GU2 7XH
{r.duangsoithong,t.windeatt}@surrey.ac.uk

Abstract. Small number of samples with high dimensional feature space leads to degradation of classifier performance for machine learning, statistics and data mining systems. This paper presents a bootstrap feature selection for ensemble classifiers to deal with this problem and compares with traditional feature selection for ensemble (select optimal features from whole dataset before bootstrap selected data). Four base classifiers: Multilayer Perceptron, Support Vector Machines, Naive Bayes and Decision Tree are used to evaluate the performance of UCI machine learning repository and causal discovery datasets. Bootstrap feature selection algorithm provides slightly better accuracy than traditional feature selection for ensemble classifiers.

Keywords: Bootstrap, feature selection, ensemble classifiers.

1 Introduction

Although development of computer and information technologies can improve many real-world applications, a consequence of these improvements is that a large number of databases are created especially in medical area. Clinical data usually contains hundreds or thousands of features with small sample size and leads to degradation in accuracy and efficiency of system by curse of dimensionality and over-fitting. Curse of dimensionality [1] , leads to the degradation of classifier system performance in high dimensional datasets because the more features, the more complexity, harder to train classifier and longer computational time. Over-fitting usually occurs when the number of features is high compared to the number of instances. The resulting classifier works very well with training data but very poorly on testing data.

To overcome this high dimensional feature spaces degradation problem, number of features should be reduced. There are two methods to reduce the dimension: feature extraction and feature selection. Feature extraction transforms or projects original features to fewer dimensions without using prior knowledge. Nevertheless, it lacks comprehensibility and uses all original features which may be impractical in large feature spaces. On the other hand, feature selection selects optimal feature subsets from original features by removing irrelevant and

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 28–41, 2010.

redundant features. It has the ability to reduce over-fitting, increase classification accuracy, reduce complexity, speed of computation and improve comprehensibility by preserving original semantic of datasets. Normally, clinicians prefer feature selection because of its understandbility and user acceptance.

There are many applications that applied feature selection as an important pre-processing step to improve systems efficiency, such as web text mining and e-mail classification, intrusion detection, biomedical informatics, gene selection in micro array data, medical data mining, and clinical decision support systems.

Feature selection is important whether the classifier is Multilayer Perceptron (MLP), Support Vector Machines (SVM) or any other classifier. Generally, feature selection can be divided into four categories: Filter, Wrapper, Hybrid and Embedded methods [2], [3], [4]. Filter method is independent from learning method used in the classification process and uses measurement techniques such as correlation, distance and consistency to find a good subset from entire set of features. Nevertheless, the selected subset may or may not be appropriate with the learning method. Wrapper method uses pre-determined learning algorithm to evaluate selected feature subsets that are optimum for the learning process. This method has high accuracy but is computationally expensive. Hybrid method combines advantage of both Filter and Wrapper. It evaluates features by using an independent measure to find the best subset and then uses a learning algorithm to find the final best subset. Finally, Embedded method interacts with learning algorithm but it is more efficient than Wrapper method because the filter algorithm has been built with the classifier. Example of Embedded method is Recursive Feature Elimination (RFE) that is embedded with Support Vectors Machines.

Feature selection has four basic processes [2]: Subset generation, subset evaluation, stopping criterion and subset validation. Subset generation produces

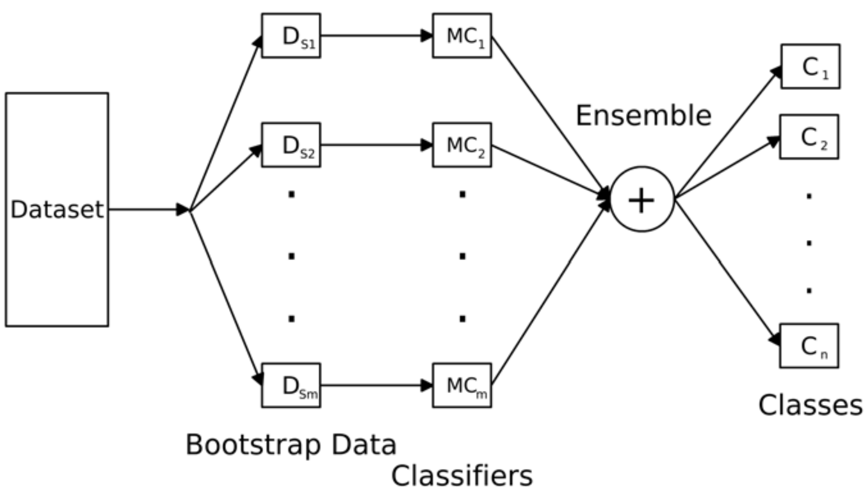

Fig. 1. Ensemble classifiers without feature selection

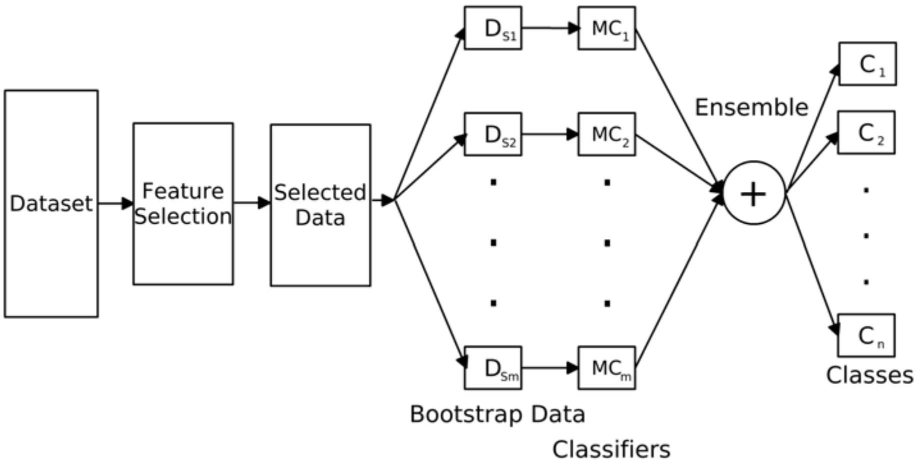

Fig. 2. Ensemble classifiers with feature selection

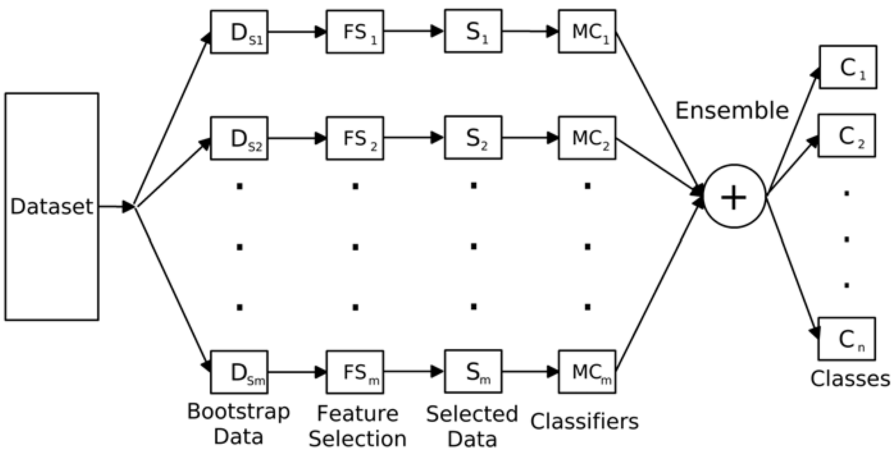

Fig. 3. Bootstrap feature selection for ensemble classifiers

candidate subset by complete, sequential or random search with three directions: forward, backward and bidirectional. After that, the candidate subset is evaluated based on criteria such as distance, dependency, information gain or consistency measurement. The process will stop when it reaches the stopping criterion. Finally, the selected subset is validated with validation data.

An ensemble classifier or multiple classifier system (MCS) is another well-known technique to improve system accuracy [5]. Ensemble combines multiple base classifiers to learn a target function and gathers their prediction together. It has ability to increase accuracy of system by combining output of multiple experts to reduce bias and variance, improve efficiency by decomposing complex problem into multiple sub problems and improve reliability by reducing uncertainty. To increase accuracy, each classifier in the ensemble should be diverse or unique in order to reduce total error such as starting with different input, initial weight, random features or random classes [6].

A typical ensemble classifier system without using feature selection is shown in Figure 1. Normally, feature selection is an essential pre-processing step to improve system performance by selecting optimal features from entire datasets as shown in Figure 2. In this paper, we present a bootstrap feature selection for ensemble classifiers as shown in Figure 3. The original dataset is divided into m bootstrap replicates and uses feature selection in order to remove redundant or irrelevant features of each bootstrap replicate. After that, selected features are passed through ensemble classifier using ensemble algorithm for training and predicting output.

The structure of the paper is the following: Introduction and related research are briefly described in section 1 and Section 2. Section 3 explains theoritical approach of feature selection, bootstrap and ensemble classifiers. The dataset and evaluation procedure are described in Section 4. Experimental results are presented in Section 5 and are discussed in Section 6. Finally, Conclusion is summarized in Section 7.

2 Related Research

Feature selection and ensemble classification have received attention from many researchers in statistics, machine learning, neural networks and data mining areas for many years. At the beginning of feature selection history, most researchers focused only on removing irrelevant features such as ReliefF [7], FOCUS [8] and Correlation-based Feature Selection(CFS) [9]. Recently, in Yu and Liu (2004) [10], Fast Correlation-Based Filter (FCBF) algorithm was proposed to remove both irrelevant and redundant features by using Symmetrical Uncertainty (SU) measurement and was successful for reducing high dimensional features while maintaining high accuracy.

According to Deisy et al. (2007) [11], SU does not have enough accuracy to quantify the dependency among features and does not take into account the effect of pairs of features on the class label during redundancy analysis. Decision Independent Correlation (DIC) and Decision Dependent Correlation (DDC) were proposed instead of using SU to remove irrelevant and redundant features, respectively. DIC and DDC provide better performance than FCBF algorithm in terms of number of selected features, computational time and accuracy.

In Chou et al. (2007) [12], modified FCBF algorithm was used to eliminate both redundant and irrelevant features for intrusion detection. In redundancy analysis, they proposed to calculate SU between features and all original features. They found that FCBF algorithm possibly keeps redundant features in the final optimal subset because it considers only SU between selected features and the rest of features at a time.

In this paper, to overcome this problem of FCBF algorithm, bootstrap feature selection for ensemble classifiers is proposed. Dataset is divided to m bootstrap replicates and selects optimal features from each bootstrap replicate. Finally, ensemble classifiers are considered by using majority vote. This algorithm also solves the small sample size problem by using bootstrap and feature selection techniques.

2.1 Feature Selection with Ensemble Classification

Although feature selection is widely used, there has been little work devoted explicitly to handling feature selection in the context of ensemble classifiers. Many previous researches have focused on determining feature subsets to combine with different ways of choosing subsets. Ho (1998) [13] presented the Random Subspace Method (RSM), one of best known feature selection with ensemble classification. It was shown that a random choice of feature subset, which allows a single feature to be in more than one subset improves performance for high dimensional problems. In Oza and Tumer (2001) [14], feature subsets are selected based on correlation between features and class. Bryll et.al. [15] presented Attribute Bagging that ranks subsets of randomly chosen features before combining. In Skurichina and Duin [16], random selection without replacement and forward features methods are used to find optimal subset. Moreover, most previous approaches have focused on determining selecting optimal features, but rarely to combine with ensemble classification.

2.2 Ensemble Feature Selection and Its Stability

In 1999, Opitz [17] proposed Genetic Ensemble Feature Selection (GEFS) algorithm by using a genetic algorithm (GA) to search and generate multiple good sets of features that are diverse from each other to use for ensemble classifiers. Asymmetric bagging of support vector machines by Li et al. [18] was proposed on predicticting drug activities for unbalanced problem between number of positive and negative samples. Munson and Caruana [19] used Bias-Variance analysis of feature selection for single and bagged model. Hybrid parallel and serial ensemble of tree-based feature selection are proposed by Tuv et al. [20] to find subset of non-redundant features after removing irrelevant features.

Y. Saeys [21] proposed a method to evaluate ensemble feature selection by measuring both stablility (robusness) and classification performance. Gulgezen et al. [22] also proposed stability measurement of MRMR (Minimum Redundancy Maximum Relevance) feature selection by using two feature selection criteria: MID (Mutual Information Difference) and MIQ (Mutual Information Quotient) and proposed new feature selection criterion MIDα.

3 Theoretical Approach

In our research, two correlation-based feature selection: Fast Correlation-Based Filter (FCBF) [10] and Correlation-based Feature Selection with Sequential Forward Floating Search (CFS+SFFS) [9],[23] are investigated for Bagging [24] ensemble classifiers, described in Section 2.2, and experimentally compared with different learning algorithms.

3.1 Feature Selection

Fast Correlation-Based Filter (FCBF). FCBF [10] algorithm is a correlation-based filter that ranks and removes irrelevant and redundant features by measuring Symmetrical Uncertainty (SU) between feature and class and between feature and feature. FCBF has two stages: relevance analysis and redundancy analysis.

Relevance Analysis. Normally, correlation is widely used to analyze relevance. In linear systems, correlation can be measured by linear correlation coefficient.

$$r = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2}\sqrt{\sum_i (y_i - \overline{y_i})^2}} \tag{1}$$

However, most systems in real world applications are non-linear. Correlation in non-linear systems can be measured by using Symmetrical Uncertainty (SU).

$$SU = 2\Big[\frac{IG(X|Y)}{H(X)H(Y)}\Big] \tag{2}$$

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

$$H(X) = -\sum_i P(x_i) log_2 P(x_i) \tag{4}$$

where $IG(X|Y)$ is the Information Gain of X after observing variable Y. $H(X)$ and $H(Y)$ are the entropy of variable X and Y, respectively. $P(x_i)$ is the probability of variable x.

SU is the modified version of Information Gain that has range between 0 and 1 and considers each feature separately (Univariate method). FCBF removes irrelevant features by ranking correlation (SU) between feature and class. If SU between feature and class equal to 1, it means that this feature is completely related to that class. On the other hand, if SU is equal to 0, the feature is irrelevant to this class.

Redundancy analysis. After ranking relevant features, FCBF eliminates redundant features from selected features based on SU between feature and class and between feature and feature. Redundant features can be defined from meaning of predominant feature and approximate Markov Blanket. In Yu and Liu (2004) [10], a feature is predominant (both relevant and non redundant feature) if it does not have any approximate Markov blanket in the current set.

Approximate Markov blanket: For two relevant features F_i and F_j $(i \neq j)$, F_j forms an approximate Markov blanket for F_i if

$$SU_{j,c} \geq SU_{i,c} \; and \; SU_{i,j} \geq SU_{i,c} \tag{5}$$

where $SU_{i,c}$ is a correlation between any feature and the class. $SU_{i,j}$ is a correlation between any pair of feature F_i and F_j $(i \neq j)$.

Correlation-based Feature Selection (CFS). CFS [9] is one of well-known techniques to rank the relevance of features by measuring correlation between features and classes and between features and other features.

Given number of features k and classes C, CFS defined relevance of features subset by using Pearson's correlation equation

$$Merit_s = \frac{kr_{kc}}{\sqrt{k + (k-1)r_{kk}}} \tag{6}$$

where $Merit_s$ is relevance of feature subset, r_{kc} is the average linear correlation coefficient between these features and classes and r_{kk} is the average linear correlation coefficient between different features.

Normally, CFS adds (forward selection) or deletes (backward selection) one feature at a time, however, in this research, we used Sequential Forward Floating Search (SFFS) as the search direction.

Sequential Forward Floating Search (SFFS). SFFS [23] is one of a classic heuristic searching method. It is a variation of bidirectional search and sequential forward search (SFS) that has dominant direction on forward search. SFFS removes features (backward elimination) after adding features (forward selection). The number of forward and backward step is not fixed but dynamically controlled depending on the criterion of the selected subset and therefore, no parameter setting is required.

3.2 Ensemble Classifier

Bagging. Bagging [24] or Bootstrap aggregating is one of the earliest, simplest and most popular for ensemble based classifiers. Bagging uses Bootstrap that randomly samples with replacement and combines with majority vote. Bootstrap is the most well-known strategy for injecting randomness to improve generalization performance in multiple classifier systems and provides out-of-bootstrap estimate for selecting classifier parameters [5]. Randomness is desirable since it increases diversity among the base classifiers, which is known to be a necessary condition for improved performance. However, there is an inevitable trade off between accuracy and diversity known as the accuracy/ diversity dilemma [5].

Bootstrap Feature Selection algorithm. The dataset is divided to m bootstrap replicates. Feature selection will select optimal features from each bootstrap replicate and selected features will be trained by base classifier. m bootstrap replicates are randomly sampled with replacement. Each bootstrap replicate contains, on average, 63.2 % of the original dataset or $(1-1/m)^m \cong 36.8$ % will be removed. Final output will be selected from majority vote from all classifiers of each bootstrap replicate. The architecture is given in Figure 2.

4 Experimental Setup

4.1 Dataset

The medical datasets used in this experiment were taken from UCI machine learning repository [25] : heart disease, hepatitis, diabetes and Parkinson's dataset and from Causality Challenge [26]: lucas and lucap datasets. The details of datasets are shown in Table 1. The missing data are replaced by mean and mode of that dataset. The causal datasets were chosen since they are high-dimension, and furthermore our ultimate goal is to apply ensemble feature selection to causality.

Table 1. Datasets

Dataset	Sample	Features	Classes	Missing Values	Data type
Heart Disease	303	13	5	Yes	Numeric (cont. and discrete)
Diabetes	768	8	2	No	Numeric (continuous)
Hepatitis	155	19	2	Yes	Numeric (cont. and discrete)
Parkinson's	195	22	2	No	Numeric (continuous)
Lucas	2000	11	2	No	Numeric (binary)
Lucap	2000	143	2	No	Numeric (binary)

Heart disease dataset was contributed by Cleveland Clinic foundation has 303 samples, 13 attributes with 138 samples presenting for heart disease class and 165 samples for absent class.

Diabetes dataset. Prima Indians Diabetes dataset was donated by John Hopkins University has 768 samples, 8 numeric features with tested positive and tested negative classes.

Hepatitis dataset was donated by G.Gong from Carnegie-Mellon University contains 155 instances, 19 attributes with live or die classes.

Parkinson's dataset. Parkinson's disease dataset is the speech signals recorded by Max Little from University of Oxford collaborated with the National Centre for Voice and Speech, Denver, Colorado. It has 197 samples, 23 features with two classes (healthy and Parkinson's patient).

Lucas dataset. Lucas (LUng CAncer Simple set) dataset is toy data generated artificially by causal Bayesian networks with binary features. This dataset is modeling a medical application for the diagnosis, prevention and cure of lung cancer. Lucas has 11 features with binary classes and 2000 samples.

Lucap dataset. Lucap (LUng CAncer set with Probes) is Lucas dataset with probes which are generated from some functions plus some noise of subsets of the real variables. Lucap has 143 features, 2000 samples and binary classes.

4.2 Evaluation

To evaluate the feature selection process, we use four widely used classifiers: Naive-Bayes(NB), Multilayer Perceptron (MLP), Support Vector Machines (SVM) and Decision Trees (DT). The parameters of each classifier were chosen based on the base classifier accuracy. MLP has one hidden layer with 16 hidden nodes, learning rate 0.2, momentum 0.3, 500 iterations and uses backpropagation algorithm with sigmoid transfer function. SVMs uses polynomial kernel with exponent 2 and set the regularization value to 0.7 and Decision Trees use pruned C4.5 algorithm. The number of classifiers in Bagging is varied from 1, 5, 10, 25 to 50 classifiers. The threshold value of FCBF algorithm in our research is set at zero for heart disease, diabetes, parkinson's and lucas and 0.13 and 0.15 for hepatitis and lucap dataset, respectively.

The classifier results were validated by 10 fold cross validation with 10 repetitions for each experiment and evaluated by average percent of test set accuracy.

5 Experimental Result

Table 2 shows the average number of selected features for each dataset. Figure 4 and 5 present example of the average accuracy of heart disease and lucap dataset. Y-axis presents the average percent accuracy of the four base classifiers and X-axis shows the number of ensemble from 1 to 50 classifiers. Solid line presents original data set, dashed line is the result of bootstrap feature selection using FCBF and bootstrap feature selection using CFS+SFFS is shown as short-dashed line. FCBF before Bootstrap result is presented in dotted line and CFS+SFFS before bootstrap is shown in dashed with dotted line.

Figure 6 and 7 show the average accuracy of six datasets for each classifier and average of all classifiers for all six datasets, respectively. Finally, Table 3

Table 2. Average number of selected features

Dataset (Original features)	Algorithm	Average number of selected features from bootstrap					
		Whole features	1	5	10	25	50
Heart Disease(13)	FCBF	6.00	4.00	5.60	5.70	5.96	6.08
	CFS+SFFS	9.00	7.00	7.60	7.80	7.88	7.90
Diabetes(8)	FCBF	4.00	4.00	3.80	3.80	3.80	3.84
	CFS+SFFS	4.00	5.00	4.80	4.50	4.32	4.40
Hepatitis(19)	FCBF	3.00	3.00	3.40	3.40	3.20	3.38
	CFS+SFFS	10.00	5.00	6.60	7.10	7.24	7.48
Parkinson's(23)	FCBF	5.00	5.00	4.00	4.30	4.28	4.36
	CFS+SFFS	10.00	10.00	8.20	8.00	8.12	8.18
Lucas(11)	FCBF	3.00	3.00	3.00	3.00	3.08	3.10
	CFS+SFFS	3.00	3.00	3.20	3.40	3.52	3.50
Lucap(143)	FCBF	7.00	7.00	6.60	8.60	7.88	7.94
	CFS+SFFS	36.00	36.00	32.60	33.40	33.32	32.96

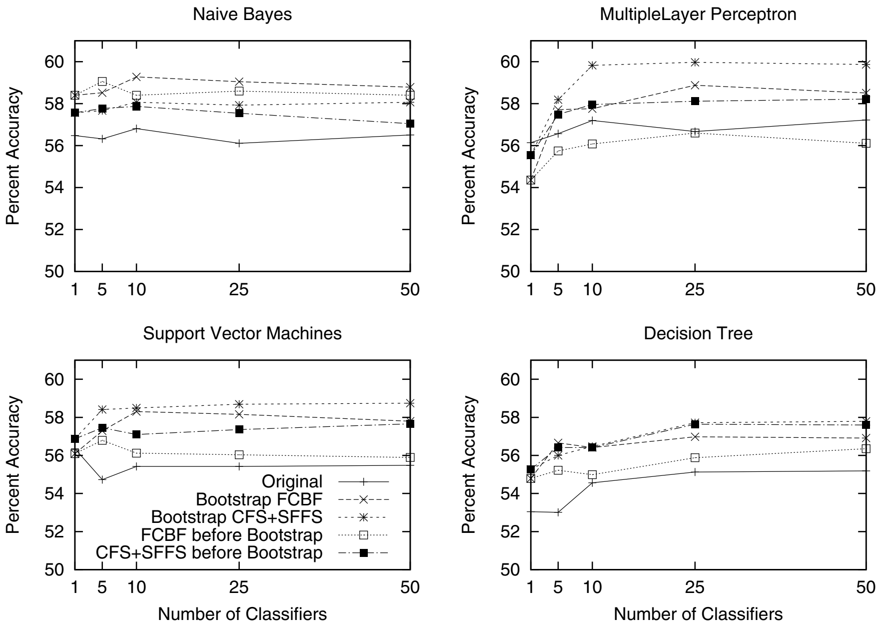

Fig. 4. Average Percent Accuracy of heart disease dataset

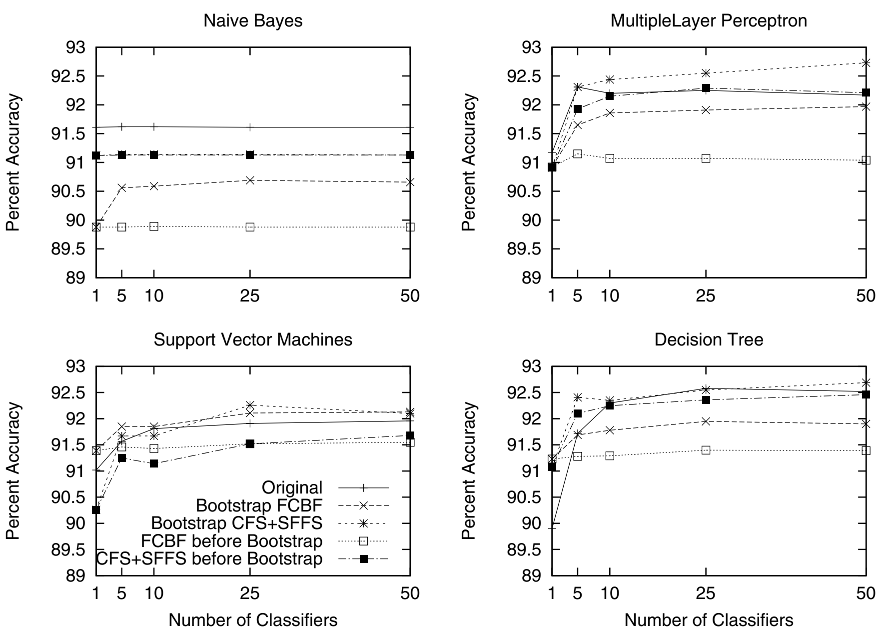

Fig. 5. Average Percent Accuracy of lucap dataset

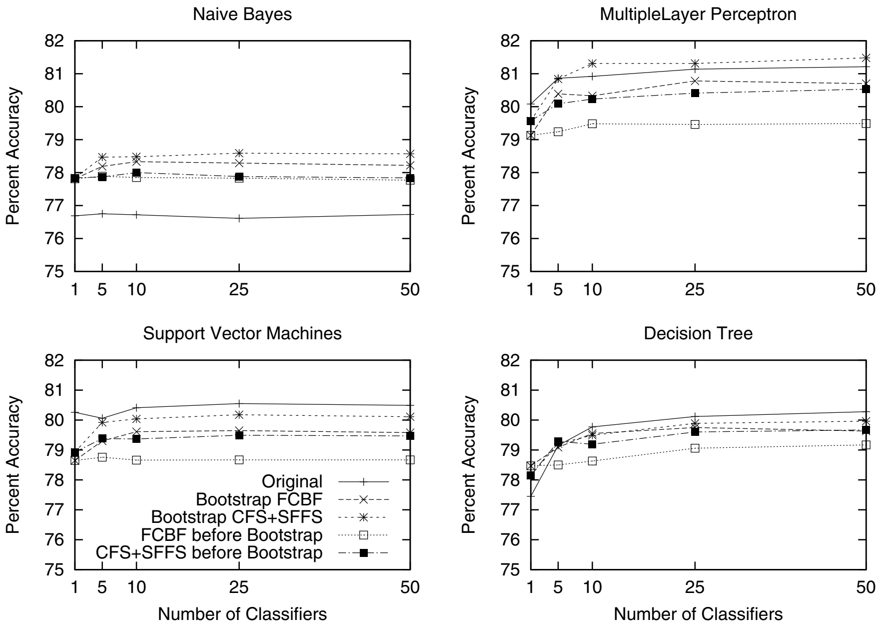

Fig. 6. Average Percent Accuracy of six datasets for each classifier

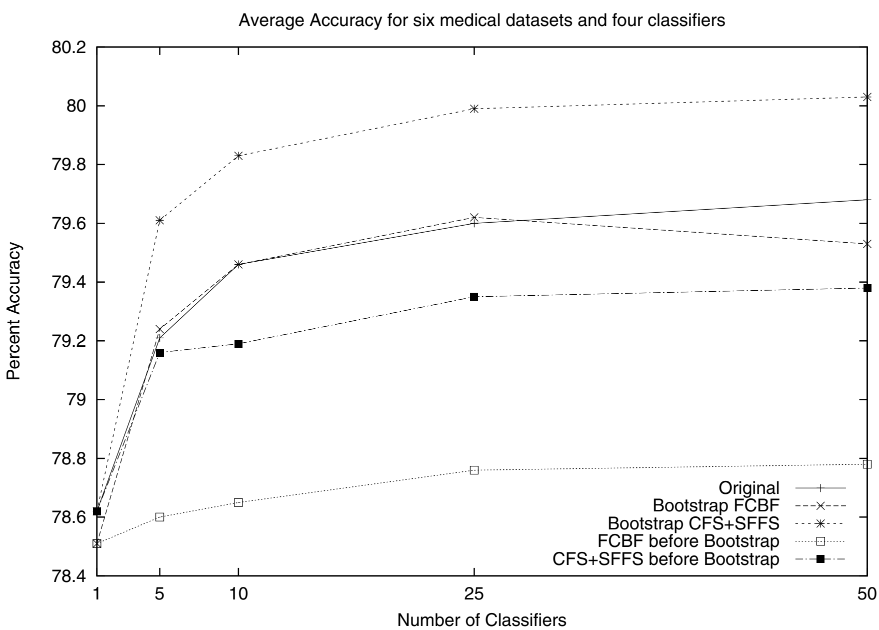

Fig. 7. Average Percent Accuracy of six datasets, four classifiers

Table 3. T-statistic test for 50 MLP classifiers of heart disease dataset

T-test		Original	bootstrap feature selection		feature selection before bootstrap	
			FCBF	CFS+SFFS	FCBF	CFS+SFFS
Original		-	1	1	0	1
Bootstrap feature selection	FCBF	0	-	1	0	0
	CFS+SFFS	0	0	-	0	0
Feature Selection before bootstrap	FCBF	1	1	1	-	1
	CFS+SFFS	0	1	1	0	-
Note: 1 = column did score significant win with regard to row						
Note: 0 = column did not score significant win with regard to row						

presents the example of T-statistic test (T-Test) for heart disease dataset using 50 MLP classifiers (the number of significant win of column compared to row).

6 Discussion

According to figures 4-7, bootstrap feature selection (figure 3) provides slightly better average accuracy than traditional feature selection for ensemble (figure 2 - feature selection from whole dataset before bootstrap selected data) in both FCBF and CFS+SFFS algorithms. On average over four classfiers and six datasets, figure 7 shows that bootstrap feature selection using CFS+SFFS provides better average accuracy than original features, bootstrap feature selection using FCBF, traditional feature selection for ensemble using CFS+SFFS and traditional feature selection for ensemble using FCBF algorithm, repectively. From table 2, it can be seen that FCBF algorithm can eliminate more redundant and irrelvant features than CFS+SFFS algorithm. Note that the average number of selected features for each number of bootstrap from 1-50 bootstrap replicates are dissimilar. This means that when we random sample with replacement, the selected feature can be different for each bootstrap replicate.

From the example of T statistic test (T-Tset) in Table 3 for heart disease dataset with 50 MLP classifiers, bootstrap feature selection using CFS+SFFS significantly improves average accuracy compared to other feature selection algorithms for ensemble. Bootstrap feature selection using FCBF algorithm also significantly outperforms other feature selection algorithms except bootstrap feature selection using CFS+SFFS. Feature selection before bootstrap using FCBF algorithm does not have significant accuracy improvement compared to other algorithms.

Furthermore, bootstrap feature selection has higher complexity than traditional feature selection for ensemble because it has to select optimal features for each bootstrap replicate.

7 Conclusions

In this paper, bootstrap feature selection for ensemble classifiers is presented and compared with conventional feature selection for ensemble classifiers. According to the average results, bootstrap feature selection for ensemble classifiers provides accuracy slightly higher than the traditional feature selection for ensemble classifiers. The only drawback of this algorithm is the complexity which is increased due to selection of optimal features of each bootstrap replicate. Future work will investigate the result of bootstrap causal feature selection for ensemble classifiers.

References

1. Bellman, R.E.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
2. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
3. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
4. Duangsoithong, R., Windeatt, T.: Relevance and Redundancy Analysis for Ensemble Classifiers. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition, vol. 5632, pp. 206–220. Springer, Heidelberg (2009)
5. Windeatt, T.: Ensemble MLP Classifier Design, vol. 137, pp. 133–147. Springer, Heidelberg (2008)
6. Windeatt, T.: Accuracy/diversity and ensemble MLP classifier design. IEEE Transactions on Neural Networks 17(5), 1194–1211 (2006)
7. Witten, I.H., Frank, E.: Data Mining Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
8. Almuallim, H., Dietterich, T.G.: Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 547–552. AAAI Press, Menlo Park (1991)
9. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceeding of the 17th International Conference on Machine Learning, pp. 359–366. Morgan Kaufmann, San Francisco (2000)
10. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. 5, 1205–1224 (2004)
11. Deisy, C., Subbulakshmi, B., Baskar, S., Ramaraj, N.: Efficient dimensionality reduction approaches for feature selection. In: International Conference on Computational Intelligence and Multimedia Applications, vol. 2, pp. 121–127 (2007)
12. Chou, T., Yen, K., Luo, J., Pissinou, N., Makki, K.: Correlation-based feature selection for intrusion detection design. In: IEEE on Military Communications Conference, MILCOM 2007, pp. 1–7 (2007)
13. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
14. Oza, N.C., Tumer, K.: Input decimation ensembles: Decorrelation through dimensionality reduction. In: Proceeding of the 2nd International Workshop on Multiple Classier Systems, pp. 238–247. Springer, Heidelberg (2001)
15. Bryll, R.K., Osuna, R.G., Quek, F.K.H.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition 36(6), 1291–1302 (2003)
16. Skurichina, M., Duin, R.P.W.: Combining feature subsets in feature selection. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 165–175. Springer, Heidelberg (2005)

17. Opitz, D.W.: Feature Selection for Ensembles. In: AAAI 1999: Proceedings of the 16th National Conference on Artificial Intelligence, pp. 379–384. American Association for Artificial Intelligence, Menlo Park (1999)
18. Li, G.Z., Meng, H.H., Lu, W.C., Yang, J., Yang, M.: Asymmetric bagging and feature selection for activities prediction of drug molecules. Journal of BMC Bioinformatics 9, 1471–2105 (2008)
19. Munson, M.A., Caruana, R.: On Feature Selection, Bias-Variance, and Bagging. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS (LNAI), vol. 5782, pp. 144–159. Springer, Heidelberg (2009)
20. Tuv, E., Borisov, A., Runger, G., Torkkila, K.: Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. Journal of Machine Learning Research 10, 1341–1366 (2009)
21. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust Feature Selection Using Ensemble Feature Selection Techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 313–325. Springer, Heidelberg (2008)
22. Gulgezen, G., Cataltepe, Z., Yu, L.: Stable and Accurate Feature Selection. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5781, pp. 455–468. Springer, Heidelberg (2009)
23. Pudil, P., Novovicova, J., Kitler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters 15, 1119–1125 (1994)
24. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
25. Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://www.ics.uci.edu/mlearn/MLRepository.html`
26. Guyon, I.: Causality Workbench (2008), `http://www.causality.inf.ethz.ch/home.php`