Ensemble-Based Wrapper Methods for Feature Selection and Class Imbalance Learning

Pengyi Yang[1,3], Wei Liu[2], Bing B. Zhou[1],
Sanjay Chawla[1], and Albert Y. Zomaya[1]

[1] School of Information Technologies, University of Sydney, NSW 2006, Australia
[2] Dept of Computing and Information Systems, University of Melbourne, Australia
[3] Garvan Institute of Medical Research, Darlinghurst, NSW 2010, Australia
yangpy@it.usyd.edu.au, wei.liu@unimelb.edu.au

Abstract. The wrapper feature selection approach is useful in identifying informative feature subsets from high-dimensional datasets. Typically, an inductive algorithm "wrapped" in a search algorithm is used to evaluate the merit of the selected features. However, significant bias may be introduced when dealing with highly imbalanced dataset. That is, the selected features may favour one class while being less useful to the adverse class. In this paper, we propose an ensemble-based wrapper approach for feature selection from data with highly imbalanced class distribution. The key idea is to create multiple balanced datasets from the original imbalanced dataset via sampling, and subsequently evaluate feature subsets using an ensemble of base classifiers each trained on a balanced dataset. The proposed approach provides a unified framework that incorporates ensemble feature selection and multiple sampling in a mutually beneficial way. The experimental results indicate that, overall, features selected by the ensemble-based wrapper are significantly better than those selected by wrappers with a single inductive algorithm in imbalanced data classification.

1 Introduction

Feature selection is a critical procedure for high-dimensional data classification. The benefits of feature selection are several-fold and dependent on the applications. For creating classification models, feature selection can often improve predictive accuracy and comprehensibility [1]. For many bioinformatics applications, feature selection is a critical procedure for identifying important biomarkers [2].

The techniques for feature selection are commonly classified as filter approach, wrapper approach, and embedded approach. Filter approach and embedded approach are relatively computationally efficient and are commonly applied as a fast feature ranking procedure [3]. In contrast, wrapper approach evaluates features by performing internal classification with a given inductive algorithm [4]. Therefore, they are much more computation intensive. Nevertheless, wrapper approach remains attractive for two reasons. Firstly, wrapper approach evaluates features iteratively with respect to an inductive algorithm. Therefore, features

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 544–555, 2013.

selected by wrapper approach are more likely to suit the inductive algorithm, and therefore, yield high classification accuracy [4]. Secondly, wrapper approach evaluates features jointly and are effective in capturing intrinsic relationships such as interactions among multiple features [5].

Learning from imbalanced data is an important problem in many data mining applications. Such a case arises when samples from one class significantly outnumber those from the other class. Imbalanced data are common in text mining [6] and bioinformatics where the minority class often represents the rare cases. It is well known that many classification algorithms are sensitive to the imbalanced class distribution [7]. Therefore, many strategies have been proposed to deal with class imbalance learning. Generally, they fall into two categories: cost-sensitive learning and data sampling [8]. With cost-sensitive learning, a given algorithm will receive a higher penalty when a mistake is made on the minority class than on the majority class. The advantage of cost-sensitive learning is that it does not modify the class distribution. However, an accurate cost-metric needs to be specified beforehand. As for data sampling, the learning instances in the majority class and minority class are manipulated in certain way so as to balance the class distribution. The downside is that sampling strategies may introduce noise or remove useful information while modifying class distribution.

The challenges of feature selection and imbalanced data classification meet when the dataset to be analysed is of both high-dimensionality and highly imbalanced class distribution [9]. In such a scenario, if wrapper approach is adopted for feature selection, the inductive algorithm may introduce significant bias because the merit of the feature subset is evaluated based on the performance of the inductive algorithm. Therefore, if the inductive algorithm favours a single class, the features selected will also bias to this class while being less useful to the adverse class.

In this study, we propose an ensemble-based wrapper approach for feature selection from highly imbalanced datasets. The proposed algorithm retains the advantages of wrapper feature selection while also maximises data usage and reduce feature selection bias simultaneously by training multiple base classifiers with balanced sample subsets. A hybrid multiple sampling procedure is employed to create balanced sample subsets. Together we introduce a unified framework that incorporates ensemble feature selection and multiple sampling in a mutually beneficial manner.

The paper is organised as follows. In Section 2, we outline the proposed framework and describe each component in details. Section 3 describes the experimental procedure. Results are presented in Section 4 and Section 5 concludes the paper.

2 Ensemble-Based Wrapper Approach

Wrapper algorithms, in general, consist of three main components [10]: (1) a search algorithm, (2) a fitness function, and (3) an inductive algorithm. The proposed system adheres to this structure. In this section, we outline the system and describe each component.

2.1 System Overview

A schematic representation of the proposed ensemble-based wrapper approach is shown in Figure 1. The imbalanced training dataset is balanced by a hybrid sampling approach (which will be explained in Section 2.3). Such a hybrid sampling procedure is applied multiple times producing multiple sets of balanced training data each of which is used to train a base classifier. The base classifiers trained on the balanced datasets are subsequently applied to classify an imbalanced test dataset. The classification distributions of each sample in the test dataset are normalised and combined, and the area under ROC curve (AUC) is calculated as the fitness indices for feature selection. The wrapper procedure terminates when it reaches a predefined number of iteration or a desired number of features is selected (i.e. greedy search), and the final feature subsets are used for further validation.

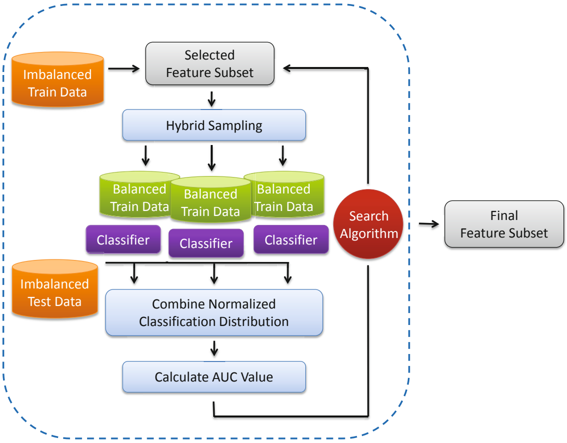

Fig. 1. A schematic representation of the ensemble-based wrapper approach

2.2 Search Algorithm

There are several popular search strategies, including hill climbing algorithms best exemplified by forward selection and backward elimination [11,12] and evolutionary algorithms such as genetic algorithm [13] and particle swarm optimisation [14].

In this study, we apply two search algorithms. The first one is a hill climbing algorithm that starts with an empty set and greedily selects a feature at a time that maximises the given fitness function. This is a typical greedy forward selection approach and at each step the best feature f^* is determined by:

$$f^* = \arg\max_{f \notin \mathbf{S}} fitness(\mathbf{S} \cup \{f\})$$

where $\mathbf{S}$ is the set that contains the features selected so far and f is a feature under evaluation according to a fitness function.

The second search algorithm is a simple elitism genetic algorithm. The feature size is pre-specified and the algorithm selects the best feature set that maximises the given fitness function through genetic operations such as crossover and mutation. Here each feature in the best set $\mathbf{S}^*$ is determined simultaneously:

$$\mathbf{S}^* = \arg\max_{i=1...p} fitness(\mathbf{S}_i)$$

where p is the population size of the genetic algorithm.

The above two typical yet simple wrapper procedures offer a transparent way to compare different inductive components.

2.3 Hybrid Sampling from Imbalanced Data

Sampling is a popular approach to balance the dataset with imbalanced class distribution. The simplest methods are random under-sampling and random over-sampling [15]. The random under-sampling method balances the dataset by randomly removing samples in the majority class. On the contrary, the random over-sampling method balances the dataset by sampling from the minority class with/without replacement and reattaching them to the dataset. A more sophisticated approach is to synthesise "new" samples from the minority class (known as SMOTE) [16]. Several studies also found that better results can be achieved by increasing minority samples and decreasing majority samples simultaneously [17,18].

Here we apply our own hybrid approach in which the dataset (denoted as $\mathbf{D}$) is balanced by increasing minority class with SMOTE and decreasing majority class with random under-sampling as follows:

$$\begin{aligned} &I_R = Random(I_{maj},\ (N_{maj} - 3/2 \times N_{min})) \\ &I_S = SMOTE(I_{min},\ 1/2 \times N_{min}) \\ &\mathbf{D}^* = (I_{min} \cup I_S) \cup (I_{maj} \backslash I_R) \end{aligned}$$

where I_{maj}, I_{min}, N_{maj} and N_{min} are the majority samples, minority samples, and their sample sizes, respectively. $Random(.)$ randomly selects from I_{maj} a subset of samples I_R and $SMOTE(.)$ creates synthetic samples I_S using I_{min}. The balanced dataset $\mathbf{D}^*$ retains the original minority samples and introduces $1/2 \times N_{min}$ synthetic minority samples. The majority samples I_R are reduced to match the new set of minority samples in $\mathbf{D}^*$ and result in a class ratio of 1.

2.4 Ensemble Learning

The classic idea of ensemble is to generate multiple datasets using a sampling method such as bootstrap, and train a set of homogeneous learning algorithms which classify new instances in a consensus manner [19]. This idea has been extended both to imbalanced data classification [20] and feature filtering [21]. However, no work has been done to unify them as a single procedure which may be mutually beneficial.

Here, we extend the idea of ensemble to feature selection in a wrapper manner and provide a unified framework that incorporates ensemble feature selection as well as multiple sampling. Specifically, given a training dataset constrained by a set of features $\mathbf{S}$, suppose we apply the above hybrid sampling procedure L times, each time producing a balanced sampling dataset $\mathbf{D}_i^{*\mathbf{S}}$ $(i = 1...L)$, and each sampling dataset is used to train a base classifier denoted as h_i. Then, the ensemble classification distribution y of each test sample $\mathbf{x}$ is computed as follows:

$$p^E(y|\mathbf{x}, \mathbf{S}) = \frac{1}{L}\sum_{i=1}^{L} Prob(h_i(\mathbf{x}), \mathbf{D}_i^{*\mathbf{S}})$$

where $Prob(h_i(\mathbf{x}), \mathbf{D}_i^{*\mathbf{S}})$ is a probability vector computed by using an ensemble of L base classifiers (h_i), each is trained on a balanced sampling set $\mathbf{D}_i^{*\mathbf{S}}$ selected by the feature set $\mathbf{S}$. Therefore, both feature set information and data sampling information are incorporated in an ensemble framework.

2.5 Fitness Function

An inductive algorithm (classifier) is commonly used to generate fitness indices in wrapper algorithms. It is well known that the overall accuracy as a metric is biased when the class distribution is imbalanced in the data. A more reliable way to compute the fitness of a feature set in such a case is to use the area under the ROC curve (AUC). AUC is a numeric value summarising the trade-off between the true positive rate and the false positive rate across the entire sample classification distribution of a dataset.

When using a single inductive algorithm, the AUC value is directly calculated by sorting the classification probability of each sample, calculating trade-off value of the true positive rate and false positive rate at each classification threshold, and calculating the area under the trade-off values. As to the ensemble classifier, classification distribution of each sample is combined and normalised across all base classifiers. Then, the same procedure as those for a single inductive algorithm is applied to calculate the AUC value.

Accordingly, we define the fitness of a feature subset as follows:

$$fitness(\mathbf{S}) = AUC(p(y|\mathbf{x}_1, \mathbf{S})...p(y|\mathbf{x}_m, \mathbf{S}))$$

where $\mathbf{x}_1$ is the first sample in the test dataset and m is the total sample size. Function $AUC(.)$ calculates the AUC value.

2.6 Main Algorithm of Ensemble-Based Wrapper Approach

Algorithm 1 represents the core of the ensemble-based wrapper approach in pseudo-code:

Algorithm 1. Ensemble Component

```
Input: A feature subset $\mathbf{S}$; Imbalanced training set $\mathbf{D}_T$ and test set $\mathbf{D}_t$
Output: Fitness of $\mathbf{S}$
$Fit = 0$;
// constrain the data dimension using the input feature subset:
$\mathbf{D}_T^{\mathbf{S}}$ = constrainDataDimension($\mathbf{D}_T$, $\mathbf{S}$);
$E = \emptyset$;
for $i = 1$ to $L$ do
    // Sampling to create a balanced dataset using training set:
    $\mathbf{D}_i^{*\mathbf{S}}$ = hybridSampling($\mathbf{D}_T^{\mathbf{S}}$);
    // Train a base classifier using balanced dataset:
    $h_i$ = trainClassifier($\mathbf{D}_i^{*\mathbf{S}}$);
    // Add the base classifier to the ensemble:
    $\mathbf{E} = \mathbf{E} \cup h_i$;
end for
$\mathbf{D}_t^{\mathbf{S}}$ = constrainDataDimension($\mathbf{D}_t$, $\mathbf{S}$);
// Apply the ensemble of classifiers to the test set:
$Fit$ = calculateAUC($\mathbf{E}$, $\mathbf{D}_t^{\mathbf{S}}$);
return $Fit$;
```

The ensemble component is independent from the search algorithm. It is flexible and can be reused in different wrapper algorithms.

3 Experimental Procedure

In this section, we summarise the datasets used for evaluation and detail the algorithms and parameter settings. Following that, the performance evaluation is described.

3.1 Datasets and Data Partitioning

We used 5 datasets with high-dimensionality and highly imbalanced class distribution. Table 1 summarises the datasets.

Specifically, fbis, re0, and oh5 are text mining datasets extracted by Han and Karypis [22]. The ALL (acute lymphoblastic leukemia) dataset is from a leukemia study [23], and the oil dataset is from study [24].

For datasets with multiple classes, we reserved the class with the smallest number of samples as the minority class and combined the other classes as the majority class. To make the problem computationally less demanding,

Table 1. Summary of datasets

Name	# Sample	# Feature	Minority class ratio
fbis	1250	2000	0.0304
re0	1504	2886	0.0073
oh5	918	3012	0.0643
ALL	248	12626	0.0605
oil	937	50	0.0438

for datasets with very high dimensions, we applied a χ^2 filtering to reduce the feature size to 500.

The datasets are partitioned using the double-level cross-validation strategy. That is for each dataset, we partitioned it using a 2-fold stratified cross-validation to obtain the training and evaluation sets. For the training set, it is further partitioned using a 5-fold stratified cross-validation to obtain the internal training and internal testing sets for feature selection. The evaluation set is reserved from the feature selection procedure and is only used for evaluating the usefulness of the selected features after the feature selection procedure.

3.2 Algorithms and Parameter Settings

For the greedy forward feature selection algorithm, we specified it to search 20 steps in which 1 to 20 features are selected one after an other. As for the genetic algorithm, we set both the population size and the termination generation to 20. The crossover probability and the mutation probability are 0.7 and 0.1, respectively. The "chromosome" is coded as a string of feature indexes, and the chromosome size of 1 to 20 are tested which corresponds to the feature subset size of 1 to 20. Different from the greedy forward feature selection algorithm which builds the feature subset on previously selected features, the genetic algorithm tests different size of feature subsets separately.

The decision tree algorithm (J48) is used for induction in our wrapper algorithms. In ensemble learning, the decision tree algorithm is prevailingly used as the base classifier because it is relatively fast to train and unstable to small changes in the data [25]. These are the important merits to our wrapper algorithms since we need to evaluate features using multiple classifiers in an efficient manner. Yet, it is widely known that the decision tree algorithm is sensitive to the imbalance of the data class distribution [26]. Hence, it is of both theoretical and practical interests to use decision tree in our experimental settings. For the ensemble wrapper, we used the ensemble size of 20. That is 20 different sampling dataset are produced in each iteration and 20 decision tree classifiers are trained on these sampling dataset and then used for feature selection.

To evaluate the selected features, we used 6 different classification algorithms, including random forest (RF), nearest neighbour with k=3 (3-NN), nearest neighbour with k=7 (7-NN), logistic regression (LogReg), multiple layer perceptron (MLP), and alternating decision tree (ADTree). The rationale is that if the wrapper algorithm is able to select useful features, the selected features should be able to improve the classification result regardless what type of classification algorithm is used. Therefore, evaluating a wide range of different classifiers can better reflect the genuine usefulness of the selected features.

3.3 Performance Evaluation

In this study, we focus on comparing wrapper algorithms with ensemble-based imbalanced sampling and classification component to wrapper algorithms with a single inductive algorithm. We refer to the first approach as the ensemble approach and the latter as the single approach. To summarise the performance results, the AUC values obtained from each classifier using features selected by ensemble approach and single approach are compared. If the ensemble approach yields a higher AUC value compared to the single approach, we label it as "ensemble win". Similarly, if the ensemble approach yields a lower AUC value compared to the single approach, we label it as "single win". When the AUC values from these two approaches are equal, we obtain a "tie". The comparison is conducted from feature size 1 to 20.

In addition, the Friedman test [27] is applied to evaluate the performance of each classifier. The confidence of 95% is used under the null hypothesis that the performance of each classifier is not significantly different by using the features selected by the ensemble approach and the single approach. The null hypothesis is rejected if there are significant performance difference when using features selected by ensemble approach as to single approach.

4 Results

AUC comparison of ensemble wrapper and single wrapper using greedy forward feature selection with fbis and re0 dataset are plotted in Figure 2 and Figure 3, respectively. As can be seen, the ensemble wrapper approach exhibited a better performance compared to the single wrapper approach. We summarise results in Figures 2 and 3 and the rest of the comparison across using feature sets with size from 1 to 20 in Table 2 and Table 3 (see 3.3 for details of the summarisation method). Specifically, Table 2 shows the comparison of the ensemble approach and the single approach using greedy forward selection algorithm, and Table 3 shows the comparison using genetic algorithm. It is clear that across all datasets most classifiers achieves better classifications using features selected by ensemble approach than those selected by single approach. This implies that the ensemble approach is more robust to high-dimensionality and highly imbalanced class distribution. Hence, the features selected by the ensemble approach are likely to be more useful to both the majority class and the minority class.

The greedy forward selection appears to be more sensitive to ensemble component. In most cases, the improvements are significant. In comparison, genetic algorithm based selection is less sensitive to the ensemble component, and most improvements are moderate. This may attributed to their different feature selection styles. That is, greedy forward selection builds the feature subset on

Table 2. Comparison of ensemble and single approaches using greedy forward selection

	fbis dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	17	17	20	14	13	15
Single Win	3	3	0	6	7	5
Friedman Test	0.0017 ✓	0.0017 ✓	7.74e-6 ✓	0.073	0.1797	0.0253 ✓

	Re0 dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	20	20	20	20	20	20
Single Win	0	0	0	0	0	0
Friedman Test	7.74e-6 ✓	7.74e-6 ✓	7.74e-6 ✓	7.74e-6 ✓	7.74e-6 ✓	7.74e-6 ✓

	Oh5 dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	12	16	6	15	13	17
Single Win	7	3	13	4	6	2
Tie	1	1	1	1	1	1
Friedman Test	0.251	0.0029 ✓	0.108	0.011 ✓	0.108	5.79e-4 ✓

	ALL dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	15	15	15	17	15	6
Single Win	5	5	5	3	5	14
Friedman Test	0.025 ✓	0.025 ✓	0.025 ✓	0.0017 ✓	0.025 ✓	0.073

	Oil dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	19	7	10	15	18	19
Single Win	1	13	10	5	2	1
Friedman Test	5.69e-5 ✓	0.179	1	0.025 ✓	3.46e-4 ✓	5.69e-5 ✓

✓ Results with significant differences (p-value lower than 0.05) using Friedman test

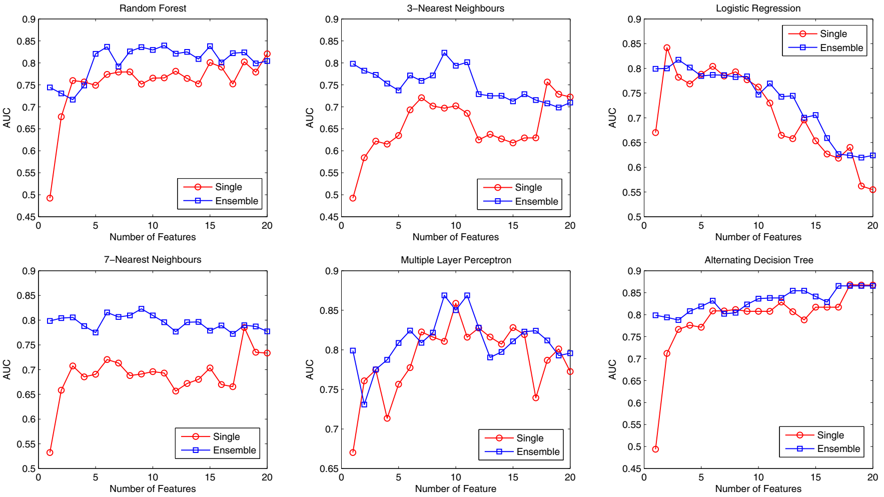

Fig. 2. AUC comparison of ensemble wrapper and single wrapper using greedy forward selection and fbis dataset. The feature size from 1 to 20 selected by ensemble and single wrappers are evaluated by 6 different classification algorithms.

Table 3. Comparison of ensemble and single approaches using genetic algorithm

	fbis dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	16	12	16	13	11	17
Single Win	4	8	4	7	9	3
Friedman Test	0.0073 ✓	0.3711	0.0073 ✓	0.1797	0.654	0.0017 ✓
	Re0 dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	13	16	18	15	11	15
Single Win	7	4	2	5	9	5
Friedman Test	0.1797	0.0073 ✓	3.46e-4 ✓	0.025 ✓	0.654	0.025 ✓
	Oh5 dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	11	12	12	14	12	11
Single Win	9	8	8	6	8	9
Friedman Test	0.654	0.3711	0.3711	0.1797	0.3711	0.654
	ALL dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	12	16	16	13	17	16
Single Win	8	4	4	7	3	4
Friedman Test	0.3711	0.0073 ✓	0.0073 ✓	0.1797	0.0017 ✓	0.0073 ✓
	Oil dataset					
	RF	3-NN	7-NN	LogReg	MLP	ADTree
Ensemble Win	12	15	12	13	19	15
Single Win	8	5	8	7	1	5
Friedman Test	0.3711	0.025 ✓	0.3711	0.179	5.69e-5 ✓	0.025 ✓

✓ Results with significant differences (p-value lower than 0.05) using Friedman test.

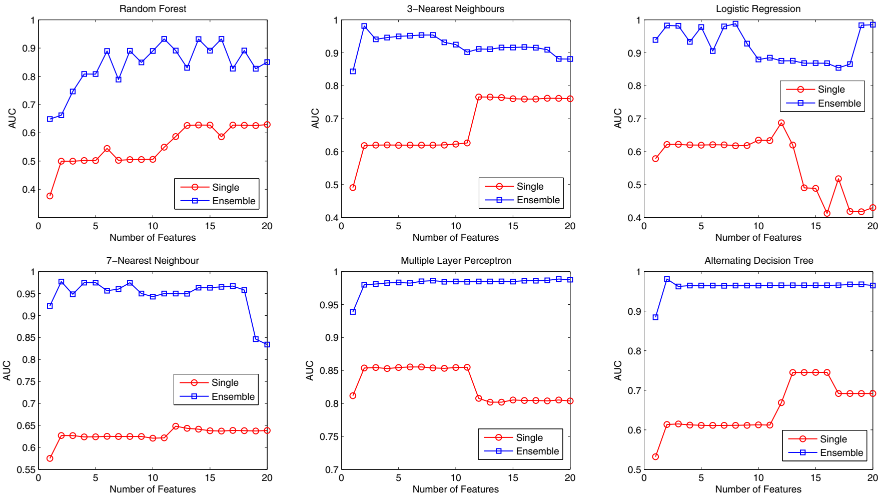

Fig. 3. AUC comparison of ensemble wrapper and single wrapper using greedy forward selection and re0 dataset. The feature size from 1 to 20 selected by ensemble and single wrappers are evaluated by 6 different classification algorithms.

previously selected features. Therefore, if a good feature is selected, it will continually be used in later iterations. Whereas, the genetic algorithm tries different size of feature subsets separately, and for each run the initiation, crossover, and mutation operations introduces randomness to the selection procedure. It follows that the greedy forward selection approach is likely to aggregate the effect of the ensemble through iterations, while the genetic algorithm approach may reduce the effect of the ensemble due to its stochastic behaviour.

It is interesting to see that different classification algorithms performed differently even with the same set of features. For the extreme case, in Table 2 the classification results of 7-NN on oh5 dataset and 3-NN and 7-NN on oil dataset contradict to the rest of the classifiers. Even for classifiers with similar comparison results, each of them may still behave differently throughout the feature subset size of 1 to 20. For example, in Figure 2, RF shows an increasing trend when more features are added. However, LogReg and ADTree indicate a decreasing trend when more features are included, whereas the performance of MLP increases first and then decreases. Note that the same sets of features and the same evaluation dataset are used for each classification algorithm. Therefore, it is clear that using a single classification algorithm for results evaluation is insufficient. Instead, multiple classification algorithms should be evaluated in order to reflect the general usefulness of the selected features.

5 Conclusion

In this study, we proposed an ensemble approach that incorporate feature selection and imbalanced data sampling in a wrapper framework. Using two search algorithms and several high-dimensional and highly imbalanced datasets, we demonstrated that features selected by the ensemble-based wrapper approach are more useful than the traditional approach (i.e. using single inductive algorithm) in terms of feature selection and imbalance learning. This implies that the traditional approach that uses a single inductive algorithm for feature evaluation may perform suboptimally when the dataset is of both high-dimensionality and highly imbalanced class distribution. By designing a multiple sampling and an ensemble feature evaluation components, we can correct the undesirable bias and identify more useful features and/or feature subsets.

References

1. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 491–502 (2005)
2. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
3. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2), 245–271 (1997)
4. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
5. Freitas, A.: Understanding the crucial role of attribute interaction in data mining. Artificial Intelligence Review 16(3), 177–199 (2001)
6. Tang, L., Liu, H.: Bias analysis in text classification for highly skewed data. In: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 784–787 (2005)

7. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 1263–1284 (2008)
8. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6(1), 20–29 (2004)
9. Mladenic, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive bayes. In: Proceedings of the Sixteenth International Conference on Machine Learning, pp. 258–267 (1999)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
11. Caruana, R., Freitag, D.: Greedy attribute selection. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 28–36 (1994)
12. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern Recognition 33(1), 25–41 (2000)
13. Oh, I., Lee, J., Moon, B.: Hybrid genetic algorithms for feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1424–1437 (2004)
14. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28(4), 459–471 (2007)
15. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis 6(5), 429–449 (2002)
16. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16(1), 321–357 (2002)
17. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Computational Intelligence 20(1), 18–36 (2004)
18. Khoshgoftaar, T., Seiffert, C., Van Hulse, J.: Hybrid Sampling for Imbalanced Data. In: Proceedings of IRI, pp. 202–207 (2008)
19. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
20. Li, C.: Classifying imbalanced data using a bagging ensemble variation (BEV). In: Proceedings of the 45th Annual Southeast Regional Conference, pp. 203–208 (2007)
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. Han, E.-H(S.), Karypis, G.: Centroid-Based Document Classification: Analysis and Experimental Results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
23. Yeoh, E., Ross, M., Shurtleff, S., Williams, W., Patel, D., Mahfouz, R., Behm, F., Raimondi, S., Relling, M., Patel, A., et al.: Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2), 133–143 (2002)
24. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30(2), 195–215 (1998)
25. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40(2), 139–157 (2000)
26. Liu, W., Chawla, S., Cieslak, D., Chawla, N.: A robust decision tree algorithms for imbalanced data sets. In: Proceedings SIAM International Conference on Data Mining, pp. 766–777 (2010)
27. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)