2014 IEEE 14th International Conference on Bioinformatics and Bioengineering

Evaluation of Wrapper-based Feature Selection using Hard, Moderate, and Easy Bioinformatics Data

Ahmad Abu Shanab, Taghi M. Khoshtoftaar, and Randall Wald
Florida Atlantic University, Boca Raton, Florida, 33431
Email: {aabusha, khoshgof, rwald1}@fau.edu

***Abstract*—One of the most challenging problems encountered when analyzing real-world gene expression datasets is high dimensionality (overabundance of features/attributes). This large number of features can lead to suboptimal classification performance and increased computation time. Feature selection, whereby only a subset of the original features are used for building a classification model, is the most commonly used technique to counter high dimensionality. One category of feature selection called wrapper-based techniques employ a classifier to directly find the subset of features which performs best. Unfortunately, noise can negatively impact the effectiveness of data mining techniques and subsequently lead to suboptimal results. Class noise in particular has a detrimental effect on the classification performance, making datasets perform poorly across a wide range of classifiers (i.e. having a high "difficulty-of-learning."). No previous work has examined the effectiveness of wrapper-based feature selection when learning from real-world high dimensional gene expression datasets in the context of difficulty-of-learning due to noise. To study this effectiveness, we perform experiments using ten gene expression datasets which was first determined to be easy-to-learn-from then had artificial class noise injected in a controlled fashion creating three levels of difficulty-of-learning (Easy, Moderate, and Hard). Using the Naïve Bayes learner, we perform wrapper feature selection followed by classification, using four classifiers (Naïve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, and Support Vector Machines), and we compare these results to the classification performance without feature selection. The results show that wrapper-based feature selection effectiveness depends on the choice of learner: for Multilayer Perceptron, wrapper selection improved performance compared to not using feature selection, while for Naïve Bayes it slightly reduced performance and for the remaining learners it further reduced performance. Because its performance relative to no feature selection varied depending on the choice of learner, we recommend that wrapper selection be at least considered in future bioinformatics experiments, especially if the goal is gene discovery not classification. Also, as dimensionality reduction techniques are not only useful but necessary for high-dimensional bioinformatics datasets, the no-feature-selection case may not be feasible in practice.**

***Keywords*-Noise injection; difficulty of learning; wrapper-based feature selection; bioinformatics**

I. Introduction

High dimensionality is one of the common characteristics exhibited by many real-world gene expression datasets. High dimensionality refers to datasets where a large number of features describe each instance, with the number of features sometimes exceeding the number of instances. This large number of features makes the analysis of such datasets more challenging, as in general, most of these features will be irrelevant to the problem at hand. The process of eliminating irrelevant features is known as feature selection, which can lead to better performance by reducing computation time, increasing the prediction accuracy of inductive models, and improving model interpretability. There are two broad categories of feature selection: filter-based techniques and wrapper-based techniques. Filter-based techniques select a feature subset using different statistical measures without involving any learning algorithm. Wrapper-based techniques use the feedback from a classifier to evaluate subsets of features. This classifier is usually the same one which will be used for building the final model. For researchers and practitioners in bioinformatics a small set of features or genes is very desirable or required. Wrapper feature selection is particularly useful here because it provides a small set of features which are unique and not highly correlated with other features in the selected set.

Noise is another difficulty encountered in many real-world gene expression datasets. Noise refers to erroneous (incorrect or missing) values in datasets, which can occur in the dependent value (class noise) or the independent values (attribute noise). Noise can hinder data mining techniques and render the data much harder to learn from. Difficulty-of-learning [5] refers to the idea that some datasets are simply more challenging than others, which will result in poor classification performance across a wide range of classification algorithms. To the best of our knowledge no previous works have investigated the effectiveness of wrapper-based feature selection in the context of difficulty-of-learning due to noise.

In this preliminary work, we create three tiers of difficulty-of-learning (Easy, Moderate, and Hard) by injecting class noise in a controlled fashion into ten gene expression datasets which were first determined to be easy-to-learn-from. This way we avoid any validity problem related to using naturally-occurring difficulty that may have hidden effects. We performed wrapper-based feature selection using the Naïve Bayes learner to counter high dimensionality. Classification models were then built using the selected features using four commonly used classifiers (Naïve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, and Support

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.62

Vector Machines). The evaluation was carried out using the area under the ROC curve (AUC) classifier performance metric. This allowed us to discover the effectiveness of wrapper-based feature selection under different difficulty-of-learning levels.

Our results show that the effectiveness of wrapper-based feature selection varies from one learner to another. In particular, the Multilayer Perceptron learner has significantly better classification performance after wrapper selection compared to no feature selection, while the Naïve Bayes learner showed only a slight difference in the performance with all features compared to the wrapper-based features. However, for the remaining learners, wrapper-based feature selection worsened the classification performance, and for these we would not recommend this form of feature selection. Nevertheless, we still recommend exploring wrapper-based feature selection as it can sometimes help improve classification performance, and in addition it can be valuable for reasons other than improving the classification performance such as knowledge discovery (i.e. finding the most important features) and eliminating redundancy among the selected features. For these latter goals, "no feature selection" is not an option.

The remainder of this paper will be organized as follows: Section II presents related works on the topics of dimensionality, data noise, and difficulty-of-learning. Difficulty-of-learning and noise injection are described in Section III. Section IV outlines the methods used in this work, the feature selection technique, the classifiers, the performance evaluation, and the datasets. In Section V, we present our results. Finally, Section VI concludes our paper and discusses the potential for future work.

II. Related Work

Datasets characterized by high dimensionality have large number of features describing each instance, or sample (in most gene expression datasets the number of features exceeds the number of instances). This overabundance of features can worsen the performance of classification models and increase computation time, because usually, most of these features are useless for building a classification model. Feature selection is the most popular technique used to counter high dimensionality, which selects the most important features and removes irrelevant and redundant features. Reducing the number of features in a dataset can improve the classification performance of classifiers, reduce the complexity of classification models, and speed up the learning process.

Much research has been done on feature selection. A good survey on various aspects of the attribute selection problem was done by Guyon and Elisseeff [8]. This study outlined key approaches used for attribute selection, including feature construction, feature ranking, multivariate feature selection, efficient search methods, and feature validity assessment methods. The authors divide feature selection techniques into two broad categories: wrapper-based techniques and filter-based techniques. Wrapper-based techniques evaluate subsets of features using a classifier. This classifier is usually the same one which will be used for building the final model. On the other hand, filter-based techniques use different statistical measures to determine which features have the highest correlation with the class rather than using a classifier. Another comprehensive survey of feature selection techniques in bioinformatics can be found in the work of Saeys et al. [12]. Due to the computational complexity required by wrapper-based techniques and the chance of building an overfitted inductive model, most research work focus on filter-based techniques, especially in bioinformatics. We provide, to our knowledge, the first assessment of the effectiveness of wrapper-based feature selection in the context of difficulty-of-learning due to noise.

Noise is another prevalent challenge exhibited by gene expression datasets. Noise refers to incorrect values in the data, which can be divided into two types: attribute noise and class noise. Attribute noise occurs when values in the independent attributes are incorrect (for example, gene expression levels not recorded correctly), while class noise refers to incorrect values in the dependent attribute (for example, cancerous instances incorrectly classified as noncancerous). Zhu and Wu [20] examined these two types of noise and concluded that class noise has a more harmful effect on classification performance than attribute noise. A comprehensive survey on the sources, challenges, and solutions to address class noise can be found in the work of Frénay and Verleysen [7]. They concluded that many open research questions related to class noise and many avenues remain to be explored.

Others have examined the impact of noise on the stability/robustness of feature selection through the direct injection of artificial noise. Wald et al. [17] examined six feature rankers, and their chosen feature lists were compared both between clean and noise-injected data and among the multiple runs of noise injection. They showed that ReliefF was a particularly stable ranker, and that comparing either noisy vs. clean or noisy vs. noisy gave similar results in terms of which rankers performed best. Abu Shanab et al. [1] evaluated six commonly used feature rankers, and their chosen feature lists were compared under different data perturbation (noise injection, sampling, and noise injection followed by sampling) and for different feature subset sizes. They showed that although stability is an important evaluation criterion for feature ranking techniques, a feature ranker's stability is not an indicator of its performance in classification.

Although previous work has evaluated the challenges of high dimensionality and difficulty-of-learning due to noise, no previous work evaluated wrapper-based feature selection in the context of difficulty-of-learning due to noise using high dimensional gene expression datasets. Wald et al. [18]

explores the degree to which class imbalance (unequal distribution of instances between classes) and difficulty of learning (due to class noise) affect one another and the best choices of learner and feature selection. Another study, Dittman et al. [4] investigated the effects of dataset difficulty due to noise injection on the stability of feature selection. The authors showed that in general, as the dataset difficulty increases, the stability of the generated feature subsets decreases. The present study differs from this earlier work in several key ways: (1) this study injects class noise into the data (after having been determined to be relatively easy-to-learn-from) creating three learning difficulty tiers (Easy, Moderate, and Hard); (2) this study employs four classifiers that are commonly-used in the literature and (3) this study investigates the effectiveness of wrapper-based feature selection when learning from high dimensional datasets with varying difficulty levels.

III. Difficulty-of-Learning and Noise Injection

A. Difficulty-of-Learning

Difficulty-of-learning [5] refers to the idea that some datasets are simply more challenging than others (in the context of building classification models with a wide range of classification algorithms), either due to the problems of noise and imbalanced data or possibly due to other causes (such as an imprecise class boundary). Previous studies have shown that it is essential to consider the difficulty level of the datasets being used [18]. The difficulty level of the dataset is obtained by first finding the performance of six commonly-used learners: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), and two versions of C4.5 decision trees (C4.5D and C4.5N) using the Area Under the Receiver Operating Characteristic (AUC) on the raw dataset, with no noise injection or other preprocessing. Then the average AUC value across six learners is found. The datasets are categorized as Easy, Moderate, and Hard based on the average AUC and according to the following ranges: Easy(> 0.8), Moderate (≤ 0.8 and > 0.7), and Hard (≤ 0.7). Note that this process is only used to determine the difficulty-of-learning and the value found has no further effect on the experiment. In this work we created the three levels of learning difficulty by injecting 24 different class noise patterns into ten high dimensional bioinformatics datasets which was first determined to be easy-to-learn-from. Due to the fact that the three difficulty levels are created in a controlled fashion (using noise injection), we avoid any validity problems associated with analyzing dataset that are already difficult-to-learn-from. Naturally-occurring difficulty may have hidden effects that might cause validity problems. Thus, to ensure that differences between difficulty levels are only caused by the difficulty itself, and not by unknown factors, it is necessary to create this difficulty through the introduction of artificial noise.

B. Noise Injection

We used the same noise injection mechanism proposed by Van Hulse et al. [15] where class noise is injected into the training datasets using two simulation parameters. That is, the levels of class noise are regulated by two noise parameters. The first parameter, denoted α ($\alpha = 10\%, 20\%, 30\%, 40\%, 50\%$), is used to determine the overall class noise level in the data. Precisely, α is the noise level relative to the number of instances belonging to the positive class, i.e., the number of examples to be injected with noise is $2 \times \alpha \times |P|$, where $|P|$ is the number of examples in the smaller class (often referred to as the positive class). This ensures that the positive class is not drastically impacted by the level of corruption, especially if the data is highly imbalanced. The second parameter, denoted β ($\beta = 0\%, 25\%, 50\%, 75\%, 100\%$), represents the percentage of class noise injected in the positive instances and is referred to as noise distribution. Note also that because the number of instances to be corrupted is tied to the number of minority-class instances, the quantity of noise injected into the dataset can be somewhat misleading: more imbalanced datasets will be injected with less noise overall, even at higher noise levels. With five values for α and β, there are 24 different noise injection patterns (because the case with $\alpha = 50\%$ and $\beta = 100\%$ would convert all positive-class instances into negative-class instances, leaving no counterexamples to learn from).

IV. Methods

A. Wrapper Feature Selection and Performance Metric

In this study, we examine wrapper-based feature selection, since filter-based techniques are not able to consider redundancy between features. This is important because features can be useless for building a classification model for two reasons: either they contain information already represented in other features (i.e. redundant) or they do not have much correlation with the class (i.e. irrelevant). Additionally, no previous study investigated the effectiveness of wrapper-based feature selection when learning from real-world high dimensional gene expression datasets in the context of difficulty-of-learning due to noise. Wrapper-based subset selection evaluates subsets of features using a classifier. The chosen subset is used to build a classification model, and the performance of this model is then used as the score for that feature subset. In this work, we use the Naïve Bayes (discussed further in Section IV-B) classifier to build our models. We selected this learner because in practice it is a very effective classifier on a wide variety of datasets [6], as well as its relative simplicity compared to other learners (e.g., SVM and MLP). Note that Naïve Bayes was used within the wrapper regardless of the learner which would be eventually used to build the classification model; future studies will compare different learners within the wrapper step. Due to the fact that our data is imbalanced, we use the AUC (Area Under the ROC Curve) metric as the performance metric within the wrapper. The AUC (discussed further in Section IV-C) performance metric has also been proven to be statistically consistent [10]. One run of five-fold cross-validation (discussed further in Section IV-C) was used for training and testing the models used for wrapper feature selection.

As wrapper selection does not itself specify a search technique to find the feature subsets, a search algorithm must be used. Based on preliminary experimentation, we chose the Greedy Stepwise approach, which uses forward selection to build the full feature subset starting from the empty set. At each point in the process, the algorithm creates a new family of potential feature subsets by adding every feature (one at a time) to the current best-known set. The merit of all these sets are evaluated, and whichever performs best is the new best-known set. This algorithm stops when none of the new sets outperform the previous best-known set, or when a user-defined maximum number of features (in our study, 100) is reached.

B. Classifiers

Four classifiers were used in this study: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), and Support Vector Machines (SVM). These were selected due to their diversity and their prevalence in the literature. All classifiers were built using the Weka machine learning software [19], using the default parameters unless noted otherwise. Note that any changes to default parameter values were applied when experimentation showed an overall improvement of the classification performance [16].

Naïve Bayes begins with a probability model which calculates the posterior probability that the instance is the member of a specific class. The classifier would assign the instance to the class for which it has the highest posterior. Unfortunately, it is very difficult to calculate the posterior directly. Therefore it is necessary to use Bayes' rule which states that the posterior equals the ratio of the prior multiplied by the likelihood over the evidence. In reality, this process can be simplified by certain assumptions. The evidence is always constant for the specific data set and therefore can be ignored for the purposes of classification. The likelihood formula, $p(F_1, \ldots, F_n|C)$ can be simplified to $\prod_i p(F_i|C)$ due to the naive assumption that all of the features are conditionally independent of all of the other features. This naive assumption with the removal of the evidence parameter creates the Naïve Bayes classifier [19].

$$p(C|F_1, \ldots, F_n) = p(C) \prod_i p(F_i|C)$$

Multilayer Perceptron is a type of artificial neural network. Artificial neural networks consist of nodes which are arranged in sets called layers. Each node in a layer has a connection coming from every node in the layer before it and going to every node in the layer after it. Each node takes the weighted sum of all of the input nodes. Along with the weighted sums, an activation function is also applied. The application of the activation function to the result of the weighted sum allows for a more clearly defined result by further separating the instances in the two classes from each other. Neural networks are well known for being robust to redundant features. However, neural networks sometimes have problems with overfitting [9]. In this study, a network with one hidden layer consisting of three nodes was used, and the $validationSetSize$ parameter was set to "10" to leave 10% of the data aside for validating when to stop the training process.

The k-nearest neighbors, or k-NN, learner is an example of an instance based and lazy learning algorithm. Instance based algorithms use only the training data without creating statistics on which to base their hypotheses. The k-NN learner does this by calculating the distance of the test sample from every training instance, and the predicted class is derived from the k nearest neighbors. In the k-NN learner, when there is a test sample that needs to be classified, the classes for each of the k closest training samples (a k of five was used in this paper, hence the name "5-Nearest Neighbors") are tabulated and the weight of each neighbor is determined by taking a measurement of $\frac{1}{distance}$ where $distance$ is the distance from the test sample. After the classes and weights are tabulated, all of the weights from the neighbors of the positive class together and all of the weights of the negative class are added together. The prediction will be the class with the largest cumulative weight [19].

Support Vector Machines, or SVM, is one popular choice of classification algorithm. One of the most efficient ways to classify between two classes is to assume that both classes are linearly separated from each other. This assumption allows us to use a discriminant to split the instances into the two classes before looking at the distribution between the classes. A linear discriminant uses the formula $g(x|\mathbf{w}, \omega_0) = \mathbf{w}^T x + \omega_0$. In the case of the linear discriminant the only data that needs to be learned is the weight vector, $\mathbf{w}$ and the bias ω_0. One aspect that must be addressed is that there can be multiple discriminants that correctly classify the two classes. SVM is a linear discriminant classifier which assumes that the best discriminant maximizes the distance between the two classes. This is measured in the distance from the discriminant to the samples of both classes [11]. For our experiments, the complexity constant "c" was changed from 1.0 to 5.0 and the "buildLogisticModels" parameter was set to TRUE.

C. Performance Evaluation and Cross Validation

We used four runs of five-fold cross-validation [19] to build and test our models. Cross-validation allows for all instances to participate both in training and testing models, without the risk of overfitting which can come from using an instance for both at the same time. The cross-validation process begins by dividing the data into N equal-size subsets (folds), with the class imbalance level in each fold made as close as possible to the original level of imbalance. One of these folds is held aside as a test (hold-out) fold, while the remaining $N-1$ folds are combined and collectively called the training fold. On this training fold, noise is first injected according to one of the 24 noise patterns, and then models are built on this noisy training fold and evaluated on the clean test fold. Note that in addition to the noise injection, wrapper feature selection and model builiding process is performed solely on the training dataset made up of the $N - 1$ folds. The noise injection, feature selection, and model-building process is repeated N times, so that each fold is used as the hold-out fold exactly once. The value $N = 5$ was used in this paper. Once all N folds have been used as the test datasets, the results from all test datasets are integrated into a single performance value for that dataset. This overall process is performed four times to reduce bias due to a chance split.

Due to the presence of imbalanced data, we needed to measure this performance using a metric which is able to balance increasing the number of true positives (correctly-classified instances of the minority class) without incurring too many false positives (members of the majority class incorrectly classified as being in the minority class). Thus, we chose the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) metric [19]. AUC builds a graph of the True Positive Rate vs. True Negative Rate as the classifier decision threshold is varied, and then uses the area under this graph as the performance across all decision thresholds.

D. Datasets

Ten bioinformatics datasets were used in our experiments, as detailed in Table I. They are sorted based on their level of class imbalance, as presented in the % Minority Instances column. They are all binary class datasets. That is, for all the datasets, each instance is assigned one of two class labels. Note that all datasets are high dimensional (that is, they have a large number of attributes). In addition to the basic properties of each dataset, the final column presents the Average AUC for the given dataset. This is the classification performance found when using the six commonly-used learners: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), and two versions of C4.5 decision trees (C4.5D and C4.5N) on the raw datasets as presented, with no noise injection or other preprocessing. Note that all of these values are above 0.8, which is our metric for Easy datasets.

None of these "raw" datasets were directly used in our experiments, however. Our experimental data was generated by injecting 24 noise patterns into these datasets and then labeling the results as "Easy," "Moderate," or "Hard" based on their difficulty of learning. In total, these categories had 141, 64, and 35 datasets, respectively. Due to the fact that the three difficulty levels are created in a controlled fashion (using noise injection), we avoid any validity problems associated with analyzing dataset that are already difficult-to-learn-from.

V. Results

In this work, we compare the use of wrapper feature selection with no feature selection. This study employs the NB learner and the AUC performance metric inside the wrapper process. Then, final (external) classification models are built with the selected features using four different learners (NB, MLP, 5NN, and SVM) which were then evaluated using the same metric. All experiments were performed on bioinformatics data which was first determined to be easy-to-learn-from, and which then had artificial class noise injected in a controlled fashion creating three levels of difficulty-of-learning (Easy, Moderate, and Hard). We used Greedy Stepwise approach as the search algorithm, and to avoid any validity problems related to overfitting we used four runs of five-fold cross-validation to build and test our models, presenting the average values across all folds and runs. The results are presented in Tables II. Within each column, **bold** values represent the best AUC value, and *italics* values represent the worst value.

We find that whether or not wrapper selection will improve performance compared to the no-selection case will depend on the choice of learner: for the MLP learner, the performance with the selected features was much better than the performance using all features, while the NB learner showed only slightly better performance with all features than with the wrapper-based features. However, for the remaining learners, wrapper-based feature selection led to a significant decrease in classification performance, and for these we would not recommend this form of feature selection.

Nonetheless, wrapper-based feature selection is useful beyond improving classification performance: it can help reveal which features are most important, giving a much smaller list compared to the full feature list and eliminating redundant features which might clutter up the list. For our datasets, wrapper-based feature selection chose fewer than 100 features, compared to the full feature sets which contained between 6,001 and 15,155 features. Even though wrapper-based feature selection was not always useful for improving classification performance, we still recommend that it be considered because it is not obvious in advance whether a given choice of learner will benefit or suffer from the use of wrapper-based feature selection.

Name	# Minority Instances	Total # of Instances	% Minority Instances	# of Attributes	Average AUC
Ovarian Cancer [13]	91	253	35.97%	15155	0.97388
ALL AML Leukemia [13]	25	72	34.72%	7130	0.90908
CNS MAT [3]	30	90	33.33%	7130	0.83551
Prostate MAT [3]	26	89	29.21%	6001	0.90466
MLL Leukemia [13]	20	72	27.78%	12583	0.89615
Lymphoma MAT [3]	19	77	24.68%	7130	0.83659
ALL [13]	79	327	24.16%	12559	0.84748
Lung Clean	23	132	17.42%	12601	0.92351
Lung Cancer [14]	31	181	17.13%	12534	0.93885
Lung Michigan [2]	10	96	10.42%	7130	0.97384

Table I: Details of the Datasets

Learner	Difficulty-of-Learning					
	Easy		Moderate		Hard	
	No Feature Selection	Wrapper Feature Selection	No Feature Selection	Wrapper Feature Selection	No Feature Selection	Wrapper Feature Selection
NB	0.876450	*0.867986*	0.755179	*0.721312*	0.630153	0.621130
MLP	*0.784365*	**0.894154**	*0.667342*	**0.778169**	*0.599262*	**0.655581**
5-NN	0.947170	0.873441	0.868226	0.746842	0.728286	*0.613167*
SVM	**0.962189**	0.892210	**0.901720**	0.770426	**0.763046**	0.633657

Table II: Average AUC Values For The Three Difficulty Levels (Easy, Moderate, Hard)

VI. Conclusion

To the best of our knowledge this is the first study to evaluate wrapper-based feature selection when learning from high dimensional bioinformatics datasets in the context of difficulty-of-learning due to noise. We injected 24 different combinations of noise level and noise distribution into ten gene expression datasets which was first determined to be easy-to-learn-from creating three levels of difficulty-of-learning (Easy, Moderate, and Hard). We then performed wrapper feature selection using the Naïve Bayes learner internally followed by external classification using one of four learners (Na¨ve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, Support Vector Machines, and Logistic Regression). We used the area under the ROC curve (AUC) performance metric within the wrapper and to evaluate the final classification models due to the presence of imbalanced data.

Our experimental results demonstrate that the effectiveness of wrapper-based feature selection when learning from noisy high dimensional datasets depends on the choice of learner: with Multilayer Perceptron, wrapper-based feature selection is better than no feature selection, while with Naïve Bayes we find that wrapper-based feature selection is slightly worse than no feature selection, and with the remaining two learners it is meaningfully worse than no feature selection. However, because it may not be clear in advance whether wrapper feature selection will improve or worsen a given choice of learner, and due to the other benefits gained when employing wrapper feature selection (such as finding the most important features which are unique and not highly correlated with other features), we recommend exploring the use of wrapper-based feature selection for bioinformatics, especially if feature reduction (and elimination of redundant features) is more important than raw classification performance.

Future research may involve conducting more experiments, using other subset evaluation techniques as well as other learners within the wrapper beyond Naïve Bayes, examining more datasets from other application domains, and considering other preprocessing techniques that are used to counter class imbalance (e.g. data sampling).

References

[1] A. Abu Shanab, T. M. Khoshgoftaar, and R. Wald, “Impact of noise and data sampling on stability of feature selection,” *Fourth International Conference on Machine Learning and Applications*, vol. 1, pp. 172–177, 2011.

[2] D. G. Beer, S. L. R. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, M. L. Lizyness, R. Kuick, S. Hayasaka, J. M. G. Taylor, M. D. Iannettoni, M. B. Orringer, and S. Hanash, “Gene-expression profiles predict survival of patients with lung adenocarcinoma,” *Nat Med*, vol. 8, no. 8, pp. 816–824, Aug 2002. [Online]. Available: http://dx.doi.org/10.1038/nm733

[3] X.-w. Chen and M. Wasikowski, “Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems,” in *KDD ’08: Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2008, pp. 124–132.

[4] D. J. Dittman, T. Khoshgoftaar, R. Wald, and A. Napolitano, “Gene selection stability’s dependence on dataset difficulty,” in *IEEE 14th International Conference on Information Reuse and Integration (IRI 2013)*, 2013, pp. 341–348.

[5] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano, "Comparing two new gene selection ensemble approaches with the commonly-used approach," in *Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA): Health Informatics Workshop*. ICMLA, 2012, pp. 184–191.

[6] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, no. 2-3, pp. 103–130, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1007413511361

[7] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 5, pp. 845–869, May 2014.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation 2nd edition*. Prentice Hall, 1998.

[10] Y. Jiang, J. Lin, B. Cukic, and T. Menzies, "Variance analysis in software fault prediction models," in *Software Reliability Engineering, 2009. ISSRE '09. 20th International Symposium on*, Nov 2009, pp. 99–108.

[11] T.-Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *IJCBS '09: International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009.*, August 2009, pp. 517–520.

[12] Y. Saeys, I. n. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/19/2507

[13] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A comparative evaluation of feature ranking methods for high dimensional bioinformatics data," in *2011 IEEE International Conference on Information Reuse and Integration (IRI)*, August 2011, pp. 315–320.

[14] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald, "Feature selection with high-dimensional imbalanced data," in *IEEE International Conference on Data Mining Workshops, 2009. ICDMW '09.*, Y. Saygin, J. X. Yu, H. Kargupta, W. Wang, S. Ranka, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, December 2009, pp. 507–514.

[15] J. Van Hulse and T. M. Khoshgoftaar, "Knowledge discovery from imbalanced and noisy data," *Data & Knowledge Engineering*, vol. 68, no. 12, pp. 1513–1542, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169023X09001141

[16] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 935–942. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273614

[17] R. Wald, T. M. Khoshgoftaar, and A. Abu Shanab, "The effect of measurement approach and noise level on gene selection stability," in *2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2012)*, 2012, pp. 420–424.

[18] R. Wald, T. M. Khoshgoftaar, A. Fazelpour, and D. J. Dittman, "Hidden dependencies between class imbalance and difficulty of learning for bioinformatics datasets," in *IEEE 14th International Conference on Information Reuse and Integration (IRI 2013)*, 2013, pp. 232–238.

[19] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, January 2011.

[20] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, Nov 2004. [Online]. Available: http://dx.doi.org/10.1007/s10462-004-0751-8