ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Handling Class Imbalance Problem Using Feature Selection

Deepika Tiwari
Dept. of Computer Science, BIT (MESRA), Noida Campus, Noida, India

Abstract
The class imbalance problem is encountered in real-world applications of machine learning and results in a classifier's suboptimal performance. Traditional classification algorithms are difficult in dealing with imbalance data. This paper proposes a feature selection algorithm which is modified version of feature selection algorithm RELIEFF, this modifies the Original RELIEFF algorithm to handle the class imbalance problem by assigns higher weight to attributes while dealing with minority classes which results in higher weight of attributes which cater to minority samples.
The experimental results show that the proposed method obtains better performance compared to original RELIEFF algorithm.

1 Introduction

The class imbalance problem is a challenge to machine learning and data mining, and it has attracted significant research recent years. A classifier affected by the class imbalance problem for a specific data set would see strong accuracy overall but very poor performance on the minority class. The imbalance data sets are pervasive in real-world applications. Examples of these kinds of applications include biological data analysis, text classification, and image classification, web page classification among many others. The skew of an imbalanced data set can be severe. Some imbalanced data sets will only have one minority sample for every 100 majority samples. Researchers have crafted many techniques to combat the class imbalance problem, including resampling, new algorithms, and feature selection. With imbalanced data, classification rules that predict the small classes tend to be fewer and weaker than those that predict the prevalent classes; consequently, test samples belonging to the small classes are misclassified more often than those belonging to the prevalent classes. Standard classifiers usually perform poorly on imbalanced data sets because they are designed to generalize from training data and output the simplest hypothesis that best fits the data. Therefore, the simplest hypothesis pays less attention to rare cases. However, in many cases, identifying rare objects is of crucial importance; classification performances on the small classes are the main concerns in determining the property of a classification model. Why is the class imbalance problem so prevalent and difficult to overcome? First, modern classifiers assume that unseen data points on which the classifier will be asked to make a prediction are drawn from the same distribution as the training data. If testing and validation data samples were drawn from a different distribution, the trained classifier may give poor results because of the flawed model. Based on this assumption, a classifier will almost always produce poor accuracy on an imbalanced data set. In a bi-class application, the imbalanced problem is observed as one class is represented by a large amount of samples while the other is represented by only a few. The class with very few training samples and usually associated with high identification importance, is referred as the positive class; the other one as the negative class. The learning objective of this kind of data is to obtain a satisfactory identification performance on the positive (small) class. Researchers have crafted many techniques to combat the class imbalance problem, including resampling, new algorithms, and feature selection.
At the data level, the objective is to re-balance the class distribution by re-sampling the data space including oversampling instances of the positive class and under sampling instances of the negative class, sometimes, uses the combination of the two techniques. At the algorithm level, solutions try to adapt the existing classifier learning algorithms to bias towards the positive class, such as cost sensitive learning and recognition based learning. In addition to these solutions, another approach is boosting. Boosting algorithms change the underlying data distribution and apply the standard classifier learning algorithms to the revised data space iteratively. From this point of view, boosting approaches should be categorized as solutions at data level. Because the class imbalance problem is commonly accompanied by the issue of high dimensionality of the data set [9], applying feature selection techniques is a necessary course of action. Ingenious sampling techniques and algorithmic methods may not be enough to combat high dimensional class imbalance problems. In this paper, we modified the existing feature selection RELIEFF algorithm to handle the class imbalance problem. The modified algorithm assigns higher weight to attributes while dealing with samples from minority classes which results in higher weight of attributes which cater to minority samples. The experimental results show that the proposed method obtains better performance compared to other methods.
The remainder of this paper is organized as follows:
Section 2 describes related work. Section 3 describes original RELIEFF Algorithm. Section 4 gives details of modified RELIEFF algorithm. Section 5 describes the experiment details. Section 6 discusses the result. Finally, we conclude our work in Section 7 and discuss the future scopes in Section 8.

II. Related Work

A. Data Level Treatments

These techniques aim to correct problems with the distribution of a data set. Weiss and Provost noted that the original distribution of samples is sometimes not the optimal distribution to use for a given classifier [10], and different sampling techniques can modify the distribution to one that is closer to the optimal distribution.

1. Sub Sampling

Sub-sampling is the process of drawing a subset of data from a large.

2. Over Sampling

Over-sampling, in its basic form, duplicates the minority-class examples.

3. SMOTE (Synthetic Minority Over-Sampling)

This sampling technique involves artificially resampling the data set.
The end result of sampling techniques is a data set that has a balanced distribution. While these methods can result in

greatly improved results over the original data set, there are significant issues surrounding their use.

After under-sampling, classifier performance may degrade due to the potential loss of useful majority-class examples. Similarly, the additional training cases introduced by over-sampling can increase the time complexity of the classifier. In the worst case, exact copies of examples after over-sampling may lead to classifier over-fitting.

However, this is not always possible in real-world applications. In some fields, the imbalance is an inherent part of the data [10]. There are many more people living without cancer at any time than those with cancer. In other fields, the cost of the data gathering procedure limits the number of samples we can collect for use in a data set and results in an artificial imbalance [10]. Many machine learning researchers simply find that you are limited to the data you have [12].

Oversampling methods, whether they duplicate existing samples or synthetically create new samples, can cause a classifier to overfit the data [10]. While many studies have shown some benefits to artificial rebalancing schemes, many classifiers are relatively insensitive to a distribution's skew [13], so the question of whether simply modifying the class ratio of a data set will result in significant improvement is considered open by some researchers [12].

B. Algorithmic Level Treatments

A wide variety of new learning methods have been created specifically to combat the class imbalance problem. While these methods attack the problem in different ways, the goal of each is still to optimize the performance of the learning machine on unseen data. One-class learning methods aim to combat the overfitting problem that occurs with most classifiers learning from imbalanced data by approaching it from an unsupervised learning angle. A one-class learner is built to recognize samples from a given class and reject samples from other classes. These methods accomplish this goal by learning using only positive data points and no other background information. These algorithms often give a confidence level of the resemblance between unseen data points and the learned class; a classification can be made from these values by requiring a minimum threshold of similarity between a novel sample and the class [14].

The two most prominent types of one-class learners are oneclass SVMs [15] and neural networks or autoencoders [14]. Raskutti and Kowalczyk [15] showed that one-class SVMs performed better than two-class SVMs on genomic data and argued that this method generalizes well to imbalanced, high-dimensional data sets. However, Manevitz and Yousef [16] believe that the results from a classifier trained using only positive data will not be as good as the results using positive and negative data. Elkan and Noto [17] agreed and also showed that when a user has negative samples, using them instead as unlabeled samples hindered a classifier's performance. Thus, unless one has only training samples known to be from one class and no other information, oneclass learners are likely not the best approach.

Boosting schemes, similar to the AdaBoost algorithm, create a series of classifiers all applied to the same data set. The AdaBoost algorithm samples with replacement from a data set with initially equal probabilities for each sample.

After each iteration, the probabilities are updated. A sample that is correctly classified receives a lower probability to be drawn in the next iteration, and a misclassified sample receives a higher probability [18]. The result is that classifiers, further in the series, draw a training set consisting of hard-to-classify samples. Chawla et al. [19] applied boosting to the SMOTE algorithm at each step to change the weight of each sample and adjust for skew in the distribution. Sun et al. [20] used a genetic algorithm to identify the best cost matrix for an imbalanced problem and then applied a standard boosting algorithm to classify the data set.

C. Feature Selection

The curse of dimensionality tells us that if many of the features are noisy, the cost of using a classifier can be very high, and the performance may be severely hindered [10] Because the class imbalance problem is commonly accompanied by the issue of high dimensionality of the dataset [10], applying feature selection techniques is a necessary course of action. Ingenious sampling techniques and algorithmic methods may not be enough to combat highdimensional class imbalance problems. Van der Putten and van Someren [21] analyzed the data sets from the CoIL Challenge 2000 and found that feature selection was more significant to strong performance than the choice of classification algorithm and helped combat the overfitting problem the most. Forman [24] noted a similar observation on highly imbalanced text classification problems and stated

"no degree of clever induction can make up for a lack of predictive signal in the input space." Research shows that in high-dimensional data sets, feature selection alone can combat the class imbalance problem [23, 25, 24].

The goal of feature selection, in general, is to select a subset of j features that allows a classifier to reach optimal performance, where j is a user-specified parameter. For high-dimensional data sets, we use filters that score each feature independently based on a rule. Feature selection is a key step for many machine learning algorithms, especially when the data is high-dimensional. Microarray-based classification data sets often have tens of thousands of features [26], and text classification data sets using just a bag of words feature set have orders of magnitude more features than documents [24]. However, another analysis of the CoIL Challenge 2000 by Elkan [27] found that feature selection metrics were not good enough for this task; instead, the interaction between different features also needed to be considered in the selection phase. The biggest flaw that he found with most of the applied feature selection methods is that they did not consider selecting highly correlated features because they were thought to be redundant. Guyon and Elisseeff [28] gave a strong theoretical analysis as to the limits of feature selection metrics. They showed that irrelevant features on their own can be useful in conjunction with other features, and the combination of two highly correlated features can be better than either feature independently.

III. Existing RELIEFF Algorithm

The Relief algorithm is one of the very successful filtering algorithms to date.Most of the filtering methods developed in data mining and machine learning assumes conditional independence among the attributes. The Relief algorithms are very useful as they do not make the assumption that the attributes are independent of each other. These algorithms are context sensitive and can correctly estimate the quality of attributes in problems with strong dependencies among the attributes [29]. This makes Relief algorithms one of the most successful preprocessing algorithms [30].

The first version of RELIEF algorithm [31] was able to deal only

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

with binary classification problems. It could estimate the rank of an attribute whether that attribute is nominal or numerical but it could not deal with missing values of
attributes in the dataset. Later in his 1994 paper [32], Igor Kononenko proposed an algorithm, an extension of original Relief algorithm, called RELIEFF that can deal with multi-class classification problem and missing attribute values in the
dataset. The pseudo code for RELIEFF is shown below

Algorithm RELIEFF:

Input: for each training instance a vector of attribute values and the class value.

Output: the vector W of estimations of the qualities of attributes.

1. set all weights W[A] = 0.0;
2. FOR i=1 to m DO
3. randomly select an instance Ri;
4. find k nearest hits Hj;
5. FOR each class C ≠ class (Ri) DO
6. from class C find k nearest misses Mj(C);
7. END FOR;
8. FOR A=1 to a DO

$$W[A] = W[A] \sum_{j=0}^{k} diff(A, Ri, Hj)/(m.k) + \sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=0}^{k} diff\left(A, R_i, M_j(C)\right) \right] /(m.k)$$

9. END FOR;
10. END FOR;

The way that the algorithm updates the weight for each attribute is simple and intuitive.

For each attribute in randomly selected sample Ri, if that attribute has a value different for that same attribute for a sample in Hj, then the algorithm takes this as not desirable and subtracts the distance from the weight of that attribute.

The distance measures for nominal and numerical attributes are as follows,

$$diff(A, I_1, I_2) = 1 - \sum_{V}^{\#values(A)} (Pr(V|class(I_1)) \times Pr(V|class(I_2)))$$

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{\max(A) - \min(A)}$$

Similarly, for all the k-misses in Mj(C), the algorithm first takes one instance from the Mj(C) and checks whether an attribute has different value in Ri and Mj(C). If the value of the attribute is different in Ri and Mj(C), the algorithm decides this situation as desirable and calculates the distance for
that attribute and adds that distance to the weight for that attribute.

The distances are always normalized so that the weights for the attributes are always in the same range and comparable to each other. The contribution for each class of the misses is multiplied by the prior probability of that class P(C) to keep the weight in the range [0, 1] and symmetric.

The selection of k-hits and k-misses for each randomly selected sample gives robustness and more stability for RELIEFF than original RELIEF algorithm. The process of updating the weight for each attribute can be stated in the following way,

$$W[A] = Pr(diff.value\ of\ A \mid nearest\ inst.\ from\ diff.class) - Pr(diff.value\ of\ A \mid nearest\ inst.\ from\ same\ class)$$

Here Pr denotes the conditional probability.

Simply speaking, if both instances are from different class and their attribute values do not match then the algorithm increases the weight and if both instances are from same class but their attribute values are different then the algorithm decreases the weight .In case of missing values the RELIEFF algorithm modifies the 'diff' function to calculate the distance for a particular attribute. It calculates the probability that two given instances (I1, I2) have different attribute values over the class value.

For example if I1 has missing value but I2 does not then,

$$diff(A, I_1, I_2) = 1 - Pr(value(A, I_2)|class(I_1))$$

But if both I1 and I2 have missing values then the distance for attribute A is,

$$diff(A, I_1, I_2) = 1 - \sum_{V}^{\#values(A)} (Pr(V|class(I_1)) \times Pr(V|class(I_2)))$$

IV. Modified RELIEFF Algorithm

The algorithm RELIEFF is an instance-based filtering method. It means that the ability to estimate the quality of attributes by RELIEFF is highly dependent on the number of instances from different classes in the dataset. Even though, the
Algorithm is dependent on class distribution of instances; it does not consider the class distribution while ranking the attributes. For balanced datasets this filtering procedure works fine. However, the performance degrades significantly when the dataset is imbalanced or biased towards any specific class.

To compensate the degradation due to data imbalance we propose a method for calculating the weight for each attribute. In this proposed modified version of RELIEFF
, we first determine the class distribution in the training data. Then while updating the weight for each attribute, if the randomly selected instance Ri is from minority class, we put higher weight by using the following equation,

$$W[A] = W[A] \sum_{j=0}^{k} diff(A, Ri, Hj)/(m.k) + \frac{\sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=0}^{k} diff\left(A, R_i, M_j(C)\right) \left(1 + \left(1 - \frac{minor}{total}\right)\right) \right]}{m.k}$$

However, if Ri is from majority class, we keep the original weight function to estimate the weight for the attributes. In this way, we are able to handle the minority class more effectively than the original RELIEFF algorithm.

The proposed RELIEFF filtering method is stated below,

Input: for each training instance a vector of attribute values and the class value

Output: the vector W of estimations of the qualities of attributes

[1] set all weights W[A] = 0.0;

[2] FOR i=1 to m DO

[3] randomly select an instance Ri;

[4] find k nearest hits Hj;

[5] FOR each class C ≠ class (Ri) DO
[6] from class C find k nearest misses Mj(C);
[7] END FOR;
[8] FOR A=1 to a DO

$$W[A] = W[A] \sum_{j=0}^{k} diff(A, Ri, Hj) / (m.k) \; + \; \frac{\sum_{C \neq class(R_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=0}^{k} diff\left(A, R_i, M_j(C)\right) \left(1 + \left(1 - \frac{minor}{total}\right)\right) \right]}{m.k}$$

[9] END FOR;
[10] END FOR;

V. Experiment

A. Data Set

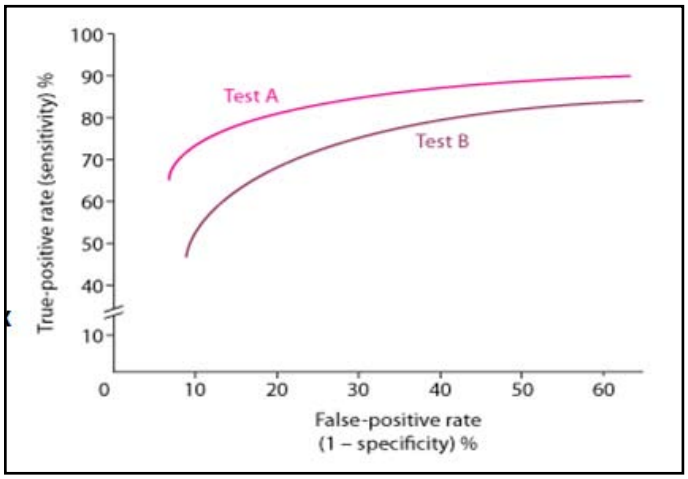
Data Set	Features	Instances	Class Ratio
Central Nervous System	7130	60	39:29
WebKb	973	5212	845:128

B. Weka

WEKA is very powerful and rich data mining software available for download free of cost from the site http://www.cs.waikato.ac.nz/ml/weka/. The standalone software provides tools for all kinds of data mining and machine learning tasks such as data pre-processing, classification, regression, clustering etc. Moreover, it is open source software written in java programming language. All the source code for different data mining and machine learning algorithms are accessible to the research community. So, researchers can easily use these algorithms and make necessary modifications suitable to any particular data mining problem.
The file format that WEKA uses for input data mining problem is called ARFF file format. This format simply defines the names of the attributes and their types followed by set of examples. Each example is a vector of the attribute values. In our research, we have modified the RELIEFF pre-processing filtering algorithm whose source code is freely available with the software itself.

C. Performance Metrics (Area under ROC Curve)

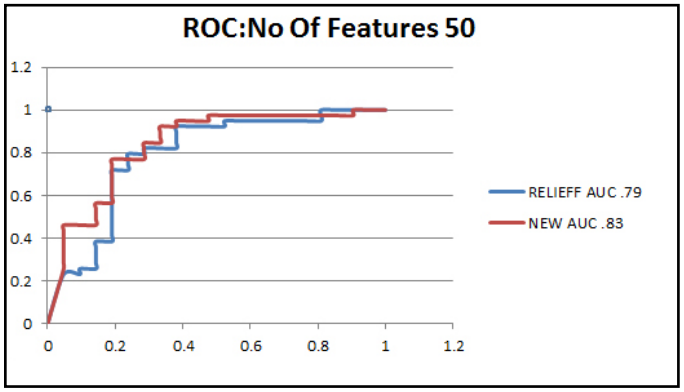
Researchers use statistics like the F-measure [24], [25] and area under the receiver operating characteristic, or ROC (AUC), [10] to better evaluate minority class performance. There are many evaluation measures in data mining, some of the most relevant ones to imbalanced datasets are: precision, recall, F-measure, Receiver Operating Characteristic (ROC) curve, cost curve and precision-recall curve. The commonality shared between them is that they are all class independent measurements. In particular, ROC curve is well received in the imbalanced dataset community and it is becoming the standard evaluation method in the area.

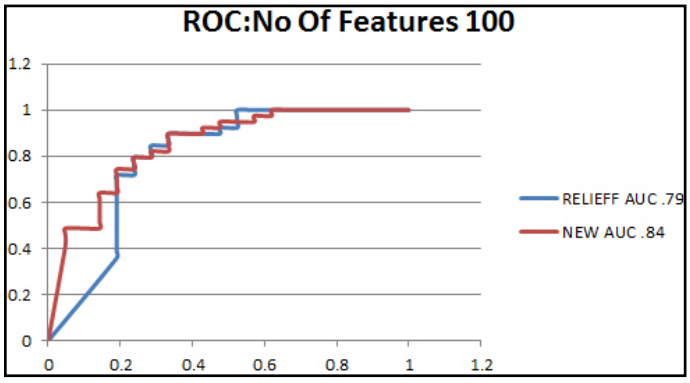


VI. Results

A. Area under ROC Curve

Features	RELIEFF	Modified Algorithm
50	.79	.83
100	.79	.84





VII. Conclusion

In this paper we modified the existing RELIEFF Feature Selection algorithm to handle the class imbalance problem. The modified algorithm shows improved performance for imbalance data set ,reducing the no of features reduces the classifier time also which results in low time & space complexity as well.

VIII. Future Work

As future work, cross validation of the experiments can be performed, and experiments can be repeated for other datasets.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

References

[1] Mike Wasikowski, Member, IEEE, and Xue-wen Chen, Senior Member, IEEE, Combating the Small Sample Class Imbalance Problem Using Feature Selection, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 10, OCTOBER 2010.

[2] Selma Ayşe Özel,A Genetic Algorithm Based Optimal Feature Selection for Web Page Classification,Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on15-18 June 2011.

[3] Yanmin Sun, Boosting for Learning Multiple Classes with Imbalanced Class Distribution,Data Mining, 2006. ICDM '06. Sixth International Conference on 18-22 Dec. 2006.

[4] David P. Williams, Member, IEEE, Vincent Myers, and Miranda Schatten Silvious,Mine Classification With Imbalanced Data,IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 6, NO. 3, JULY 2009.

[5] Nadeem Qazi,Effect Of Feature Selection, Synthetic Minority Over-sampling(SMOTE) And Under-sampling On Class imbalance Classification, 2012 14th International Conference on Modelling and Simulation.

[6] Xiaolong Zhang,Imbalanced Data Classification Algorithm Based on Boosting and Cascade Model,2012 IEEE International Conference on Systems, Man, and Cybernetics October 14-17, 2012, COEX, Seoul, Korea

[7] Sathi T Marath , Large-Scale Web Page Classification , Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at Dalhousie University Halifax, Nova Scotia November 2010.

[8] Selma Ayşe Özel Department of Computer Engineering,A Genetic Algorithm Based Optimal Feature Selection for Web Page Classification. Date of Conference: 15-18 June 2011

[9] N. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 1-6, 2004

[10] G. Weiss and F. Provost, "Learning when Training Data Are Costly: The Effect of Class Distribution on Tree Induction," J. Artificial Intelligence Research, vol. 19, pp. 315-354, 2003.

[11] X. Chen, B. Gerlach, and D. Casasent, "Pruning Support Vectorsfor Imbalanced Data Classification," Proc. Int'l Joint Conf. Neural Networks, pp. 1883-1888, 2005.

[12] Proc. AAAI '00 Workshop Learning from Imbalanced Data Sets,N. Japkowicz, ed., 2000.

[13] C. Drummond and R.C. Holte, "Exploiting the Cost (In) sensitivity of Decision Tree Splitting Criteria," Proc. 17th Int'l Conf. Machine Learning, pp. 239-246, 2000.

[14] N. Japkowicz, "Supervised versus Unsupervised Binary Learning by Feedforward Neural Networks," Machine Learning, vol. 42, nos. 1/2, pp. 97-122, 2001

[15] A. Raskutti and A. Kowalczyk, "Extreme Rebalancing for SVMs: A SVM Study," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1,pp. 60-69, 2004.

[16] L.M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," J. Machine Learning Research, vol. 2, pp. 139-154, 2001.

[17] C. Elkan and K. Noto, "Learning Classifiers from Only Positive and Unlabeled Data," Proc. ACM SIGKDD '08, pp. 213-220, 2008.

[18] E. Alpaydin, Introduction to Machine Learning, pp. 43-45, 360-363. MIT Press, 2004.

[19] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," Proc. Seventh European Conf. Principles and Practice of Knowledge Discoveryin Databases (PKDD), pp. 107-119, 2003.

[20] Y. Sun, M. Kamel, and Y. Wang, "Boosting for Learning Multiple Classes with Imbalanced Class Distribution," Proc. Sixth Int'l Conf. Data Mining, pp. 592-602, 2006.

[21] P.V. der Putten and M. van Someren, "A Bias-Variance Analysis of a Real World Learning Problem: The CoIL Challenge 2000," Machine Learning, vol. 57, nos. 1/2, pp. 177-195, 2004.

[22] X. Chen and M. Wasikowski, "FAST: A ROC-Based Feature Selection Metric for Small Samples and Imbalanced Data Classification Problems," Proc. ACM SIGKDD '08, pp. 124-133, 2008.

[23] D. Mladeni_c and M. Grobelnik, "Feature Selection for Unbalanced Class Distribution and Naive Bayes," Proc. 16th Int'l Conf. Machine Learning, pp. 258-267, 1999.

[24] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," J. Machine Learning Research, vol. 3, pp. 1289-1305, 2003.

[25] Z. Zheng, X. Wu, and R. Srihari, "Feature Selection for Text Categorization on Imbalanced Data," ACM SIGKDD Explorations Newsletter, vol. 6, pp. 80-89, 2004.

[26] H. Xiong and X. Chen, "Kernel-Based Distance Metric Learning for Microarray Data Classification," BMC Bioinformatics, vol. 7, no. 299, pp. 1-11, 2006.

[27] C. Elkan, "Magical Thinking in Data Mining: Lessons from CoIL Challenge 2000," Proc. ACM SIGKDD '01, pp. 426-431, 2001.

[28] Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," J. Machine Learning Research, vol. 3, pp. 1157- 1182, 2003.

[29] Robnik-Šikonja, M. and Kononenko, I. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF. Mach. Learn. 53, 1-2 (Oct. 2003), 23-69.

[30] Dietterich, T. G. (1997). Machine learning research: Four current directions. AI Magazine 18(4), 97--136.

[31] Kira, K. and Rendell, L. (1992), "The feature selection problem: Traditionalmethods and a new algorithm", in: Proceedings of AAAI-92, AAAI Press, 129-134.

[32] Kononenko, I. 1994. Estimating attributes: analysis and extensions of RELIEF. In Proceedings of the European Conference on Machine Learning on Machine Learning (Catania, Italy). F. Bergadano and L. De Raedt, Eds. Springer-VerlagNew York, Secaucus, NJ, 171-182.