Feature Selection for Maximizing the Area Under the ROC Curve

Rui Wang and Ke Tang
Nature Inspired Computation and Applications Laboratory (NICAL)
University of Science and Technology of China
Hefei, 230027, China
wrui1108@mail.ustc.edu.cn
ketang@ustc.edu.cn

Abstract

Feature selection is an important pre-processing step for solving classification problems. A good feature selection method may not only improve the performance of the final classifier, but also reduce the computational complexity of it. Traditionally, feature selection methods were developed to maximize the classification accuracy of a classifier. Recently, both theoretical and experimental studies revealed that a classifier with the highest accuracy might not be ideal in real-world problems. Instead, the Area Under the ROC Curve (AUC) has been suggested as the alternative metric, and many existing learning algorithms have been modified in order to seek the classifier with maximum AUC. However, little work was done to develop new feature selection methods to suit the requirement of AUC maximization. To fill this gap in the literature, we propose in this paper a novel algorithm, called AUC and Rank Correlation coefficient Optimization (ARCO) algorithm. ARCO adopts the general framework of a well-known method, namely minimal-redundancy-maximal-relevance (mRMR) criterion, but defines the terms "relevance" and "redundancy" in totally different ways. Such a modification looks trivial from the perspective of algorithmic design. Nevertheless, experimental study on four gene expression data sets showed that feature subsets obtained by ARCO resulted in classifiers with significantly larger AUC than the feature subsets obtained by mRMR. Moreover, ARCO also outperformed the Feature Assessment by Sliding Thresholds algorithm, which was recently proposed for AUC maximization, and thus the efficacy of ARCO was validated.

1. Introduction

For many years, feature selection has kept playing an important role in most data mining tasks. Given a data set represented by a number of features, feature selection aims to identify a feature subset, based on which the final model (e.g., a classifier) will be trained. Instead of training a model with all the features, conducting feature selection in prior to the training episode provides quite a few benefits [8]. First, a compact feature subset can alleviate the curse of dimensionality, and hence avoid the over-fitting scenario that is usually encountered in the training episode. Second, a model with better generalization performance may be obtained by removing the noisy features. Third, the computational cost can be significantly reduced if only a small subset of the original features is used. Finally, a descriptive feature subset can make the output of a model more explainable and understandable.

Since the reduction of computational cost can be rather easily achieved by randomly removing a number of features, the major challenge lies in seeking the feature subset that lead to comparable (or even enhanced) performance of the final model. Consequently, the efficacy of a feature selection method is commonly assessed by the performance of the final model trained with the feature subset selected (e.g., [4] [17]). In the literature, most investigations on feature selection were conducted in the context of classification, in which classification accuracy has been long believed to be the best criterion for assessing the performance of the final model (classifier). For this reason, numerous feature selection methods have been designed with respect to classification accuracy. However, it has recently been pointed out that accuracy is not always a suitable assessment metric [15]. For example, in many real practice problems, prior class distribution is imbalanced, and high accuracy can be easily archived by a trivial classifier which put every testing sample into majority class. Furthermore, making errors on different samples may incur different cost. In this case, a classifier with minimum cost is more desirable than a classifier with highest classification accuracy. However, in many real-world problems, costs associated to different types of errors are usually difficult to quantitatively define. Hence, Receiver Operating Characteristic (ROC) analysis

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.25

has emerged as an alternative metric for assessing classifiers [5]. Specifically, the Area Under the ROC Curve (AUC) has been proven to be a better performance metric in comparison with classification accuracy [11]. It is considered that the larger the AUC, the better the classifier is. Along with this new model evaluation criterion, the goal of training a model has become maximizing the AUC rather than the accuracy. Unfortunately, most of the traditional learning methods fail to produce classifiers with large AUC (which is unsurprising since they were originally developed for maximizing accuracy). Therefore, AUC maximizing variants of almost all learning methods, such as decision trees [6] and Support Vector Machines (SVM) [2], have been developed.

In the light of previous work, feature selection method with respect to AUC maximization is overdue. To our best knowledge, Chen and Wasikowski [3] might be the first who proposed a feature selection method for AUC maximization. In [3], they proposed a feature ranking method, namely Feature Assessment by Sliding Thresholds (FAST), which score each feature by calculating the AUC of this single feature classifier. Experimental study showed that FAST outperformed two commonly used feature selection approaches, feature selection with Pearson's correlation coefficients (CC) and Relief [3], in terms of AUC. However, FAST does not take into account the redundancy in the feature set. In feature selection literature, a rule of thumb is that the feature selection procedure should not only identify those features that are "good" (relevant), but also remove those redundant features [8]. Previous studies have repeatedly shown that considering both relevance and redundancy in the feature selection procedure led to better feature subset in most cases (e.g., [13]). Such experience implies that FAST can be further improved by incorporating some type of redundancy minimization strategy.

In this paper, we propose a new feature selection method, called AUC and Rank Correlation coefficient Optimization (ARCO). With this method, we aim to select a compact feature subset, which will suit the purpose of AUC maximization well. Briefly speaking, ARCO is a combination of FAST and a redundancy constraint, namely Spearman's Rank Correlation Coefficient (RCC) [12]. By combining the AUC and RCC into a single criterion, ARCO manages to both identify features with maximum AUC and ignore redundant features. Empirical comparison between ARCO and three existing methods (including FAST) clearly showed the advantage of ARCO.

The rest of this paper is organized as follows: In section 2, FAST is briefly introduced. Section 3 describes ARCO in detail. Experimental study is presented in Section 4. Finally, we conclude this paper in Section 5.

2. Feature Assessment by Sliding Thresholds

Suppose we need to select *k* features from a raw feature set $F = \{f_1, f_2, \ldots, f_m\}$. FAST views each feature as the output of a classifier, and calculates AUC for each of them. Then, the features are ranked by the AUC in descending order, and the leading *k* features are picked. The pseudo-code of FAST is as follows:

for $i = 1$ to m **do**
    auc[i]←AUC score of the f_i;
end
sort(auc);
pick out *k* features with highest auc;

Algorithm 1: FAST

Given a group of samples' score and their true class label, there are a few ways to calculate AUC. In the original paper of FAST, AUC is calculated by plotting the ROC curve and summing up the area under it. However, this approach may incur imprecise estimation of AUC [10]. Alternatively, one may employ Wilcoxon-Mann-Whitney test to calculate AUC, since they are equivalent [10]. Suppose a data set consists of *n* samples, n_0 of them are from the positive class, and the rest n_1 samples belong to the negative class. Each sample $\boldsymbol{x}_i$ is associated with a score $s(\boldsymbol{x}_i)$. To calculate AUC, these scores are first sorted in ascending order, then each of them are assigned with a rank starting from 1. After that, samples with the same score should be re-ranked by averaging the original ranks of them. Let $r_1, r_2, \ldots, r_{n_0}$ be the ranks of positive samples. AUC can be calculated using Eq. (1).

$$AUC = \frac{\sum_{i=1}^{n_0}(r_i - i)}{n_0 \times n_1} = \frac{\sum_{i=1}^{n_0}(r_i) - \frac{n_0 \times (n_0+1)}{2}}{n_0 \times n_1} \quad (1)$$

3. AUC and Rank Correlation coefficient Optimization

As mentioned in Section 1, although traditional feature selection methods might be unsuitable for AUC maximization, some useful idea can be attained from them. In particular, it is well acknowledged that redundant features should be removed by the feature selection procedure. With this in mind, we propose a new feature selection method, ARCO. ARCO combines relevance and redundancy by means of a modern and effective feature selection framework, called minimal-redundancy-maximal-relevance(mRMR) [13]. The main difference between ARCO and mRMR is their metrics of relevance and redundancy. In mRMR, mutual information (MI) between a feature and the class label is adopted to measure the relevance of this feature. The redundancy between two features is measured either by MI, or by the Pearson's correlation coefficient (CC). Consider two features f_p and f_q, the MI and

CC between them can be calculated using Eqs. (2) and (3), respectively:

$$MI(f_p, f_q) = \sum_{f_p;f_q} p(f_p, f_q) log \frac{p(f_p, f_q)}{p(f_p)p(f_q)}. \quad (2)$$

where $p(f_p)$, $p(f_q)$ are the marginal distribution of f_p and f_q. $p(f_p,f_q)$ is the joint distribution of f_p and f_q. And the sum operation run over all possible values of f_p and f_q.

$$CC(f_p, f_q) = \frac{cov(f_p, f_q)}{\sigma_{f_p}\sigma_{f_q}}. \quad (3)$$

where σ_{f_p} and σ_{f_q} are the standard deviation of f_p and f_q, and $cov(f_p,f_q)$is the covariance between these two features.

Although Eqs. (2) and (3) perform well for accuracy maximization, they are inappropriate when the goal is AUC maximization. The essential difference between accuracy maximization and AUC maximization is that the latter mainly considers whether a classifier ranks the samples correctly regardless of its ability to estimate test samples' posteriori. Neither mutual information nor Pearson's correlation coefficient is able to take care of this issue. Hence, they should be replaced by some other metrics that suit the objective of AUC maximization better. As in FAST, we also employ the AUC (of a single feature) as the relevance metric. Moreover, we propose using the RCC [12] to measure the redundancy. Given two features f_p and f_q, the RCC can be calculated as follows: First, the samples are sorted on each feature based on their values. For each sample $\mathbf{x}_i$, d_i is defined as the difference between $\mathbf{x}_i$'s ranks on the two features. Then, the RCC between f_p and f_q can be calculated by Eq. (4)

$$RCC(f_p, f_q) = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad (4)$$

RCC is a nonparametric measure based on ranks, which does not work under a special assumption about the distribution of data. In principle, RCC is equivalent to first converting the features' values into ranks, and then computing the Pearson's correlation coefficient between these ranks. Since both AUC and RCC handle ranks rather than values of the features, we expect the combination of them to work well for AUC maximization.

To select *k* features out of the whole feature set of size *m*, ARCO starts from picking out the feature with the largest AUC. After that, the rest of the features are selected iteratively. Let S be the current selected feature subset and $|S|$ be its cardinality. At each iteration, every previously unselected feature f_i is evaluated with Eq. (5), which is a combination of AUC and RCC, and the one with the largest value is selected. The pseudo-code of ARCO is presented in Algorithm 2.

$$E_i = AUC(f_i) - \frac{\left|\sum_{f_j \in S} RCC(f_i, f_j)\right|}{|S|}. \quad (5)$$

S=$\emptyset$;
calculate every feature's auc score;
put the feature with largest AUC score into S;
for $i = 2$ to k **do**
  $f' = \arg\max_{f_i \in F-S} E_i$;
  put f' into S;
end
return S

Algorithm 2: ARCO

To calculate the AUC score of a feature, one needs to sort the samples according to their values on this feature. Hence, the computational complexity of calculating the AUC score of all *m* features is *O*(*mnlogn*). This is also the complexity of FAST algorithm. To remove the redundant features, ARCO needs extra computation. Since the RCC of a pair of features can be calculated in *O*(*nlogn*) time, the complexity introduced by redundance term of ARCO is *O*(*kmnlogn*). Therefor, the complexity of ARCO algorithm is *O*(*kmnlogn*).

4. Experiments

The efficacy of ARCO was empirically evaluated by comparing it to three exiting feature selection methods on four data sets.

4.1. Data Sets

In recent years, bioinformatics has become an important application domain of data mining techniques. In particular, the gene selection problem, which aims to select features for gene expression data analysis, has attracted a lot of attention (e.g.,[9][4][17]). Furthermore, many feature selection methods, including those investigated in our experiments, has been evaluated on gene expression data sets. To facilitate the comparative study, we also used four gene expression data sets in our experimental study: colon cancer data (Colon) [1], leukemia data (Leukemia) [7], Central Nervous System Embryonal Tumor data (CNS) [14], and Lymphoma data (Lymph) [16]. All these data sets are binary class problems, with imbalanced distributed samples and considerably large numbers of features. Table 1 presents the detailed information of these data sets.

Table 1. Information of the four data sets

Data Sets	Source	Feature number	Sample number
Colon	Alon et al [1]	2000	62 (40+22)
Leukemia	Golub et al [7]	7129	72 (47+25)
CNS	Pomeroy et al [14]	7129	90 (60+30)
Lymph	Shipp et al [16]	7129	77 (58+19)

4.2. Experimental Setup

We compared ARCO with FAST, mRMR with mutual information as its relevance and redundancy metric, and Relief. To our best knowledge, FAST is the pioneer method aiming to maximize AUC, and is the one that motivated ARCO. Comparison between ARCO and FAST was to verify whether removing redundancy was still an important issue in the context of AUC maximization. mRMR provides the general framework used by ARCO. Comparing ARCO to it demonstrated the advantage of using AUC and RCC as the relevance and redundancy metrics. Relief is a classic feature selection algorithm which has been extensively studied both theoretically and empirically. Hence, it served as the baseline method in our experiments.

1-Nearest Neighbor (1NN) and Naive Bayes (NB) were used as the classification algorithm in our experiments. 1-NN is one of the simplest classification algorithms. It does not need any training process. For a given testing instance, this classifier find its nearest neighbor in training instances, then assigns this instance to the class that the training instance belongs to. Despite its vulnerable to noise and outliers, 1-NN has shown to be effective in many problems and been used frequently by researchers.

NB is another popular approach that works under the Bayes's theorem and assumes that each feature's value is unaffected by any other feature. That is:

$$p(c|f_1, \ldots, f_m) = \frac{1}{Z} p(c) \prod_{i=1}^{m} p(f_i|c) \qquad (6)$$

where c is the class variable, $p(f_i|c)$ is the conditional density learned in the training process. Z is a scaling factor to make sure the left part of this equation is in a probability form. In spite of the seemed unreasonable independence assumption, NB worked amazingly well in real world practice. As the two classification methods work in quite different ways, we expected to check whether ARCO is biased to any specific classifier.

For each data set, 100 folds bootstrap were employed to evaluate every feature selection method on it. In each bootstrap, we examined 20 groups of features with different size, from 5 to 100 with interval 5. After that, the averaged performances of each classifier with each feature selection method were calculated for each feature subset size. Then, Wilcoxon signed-rank test (significance level was set to 0.05) was employed to compare ARCO with the other three methods.

4.3. Results

We used the WEKA software package [18] to set up our experiments' platform. The four feature selection methods described above were implemented based on this platform. In Relief, every instance was used once to update the weights of features. For mRMR method, discretization was applied to the numerical features, as suggested in [4].

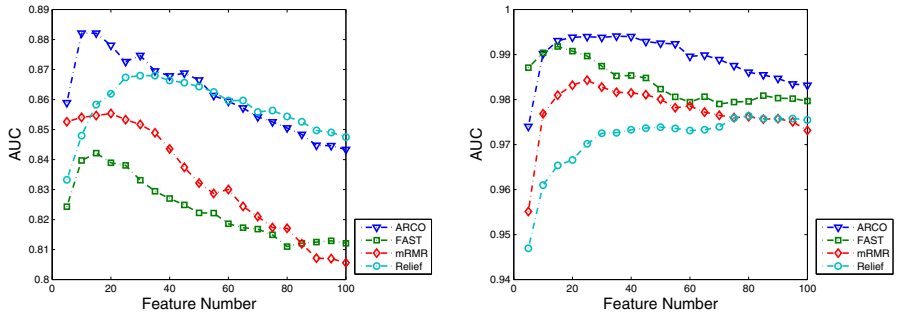
Figs. 1 and 2 show the results obtained by the four compared feature selection methods. In general, ARCO performed the best. We can observe that ARCO led to higher AUC than FAST and mRMR in most cases regardless of the classification algorithm used. When using NB, ARCO outperformed Relief on the Leukemia, CNS, and Lymph data sets, while was slightly infer on Colon data set when more than 65 features were selected. As for 1-NN, ARCO consistently outperformed Relief on all the four data sets.

Table 2 presents the results of the Wilcoxon signed-rank tests. For each cell of the table, "*w*" stands for "win", indicating the number of the cases in which ARCO was significantly better than the compared algorithm. Correspondingly, "*d*" and "*l*" (stand for "Draw" and "Lose") indicate the numbers of the cases that ARCO performed comparably or significantly worse. We can observe that ARCO performed either significantly better or comparable to the other methods in almost all the cases. To summarize, the experimental results clearly demonstrate the efficacy of ARCO as a feature selection approach for AUC maximization.

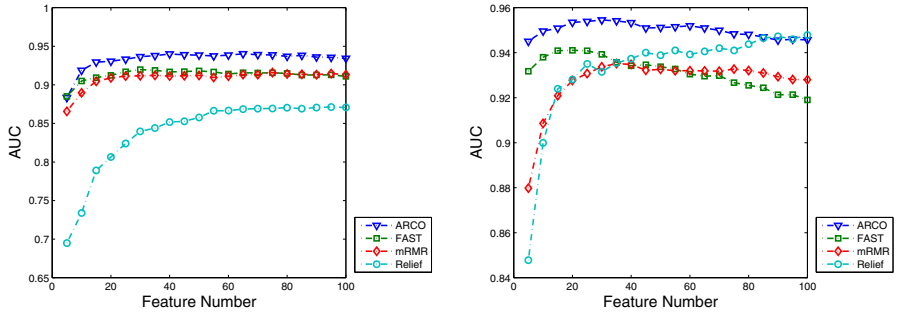
5. Conclusion

For many years, classification has stood as one of the most important tasks in data mining problems. Recently, it is gradually accepted that a classifier with large AUC is more appealing than a classifier with large classification accuracy. Hence, a lot of effort has been dedicated to make learning algorithms capable of maximizing AUC of the final classifier. As an important preprocessing step of the whole learning procedure, feature selection always affects the performance of the final classifier. However, little work has been done to design novel feature selection methods to serve the new objective of AUC maximization. This paper aims to fill this gap in the literature. As a result, the ARCO algorithm was proposed. ARCO was designed based on two previous work, the FAST and mRMR. Experimental study showed that ARCO was superior than three existing feature selection methods, i.e., FAST, mRMR, and Relief. This observation not only demonstrated the efficacy of ARCO, but also validated two hypotheses behind it. That is, the redundancy minimization is still a key issue for feature selection in the context of AUC maximization, and traditional metrics of relevance and redundancy should be replaced by some new metrics that are more closely related to the new objective.

Two directions deserve further investigation in the future. First, the efficacy of ARCO should be verified on multi-class problems. Second, we adopted very simple clas-

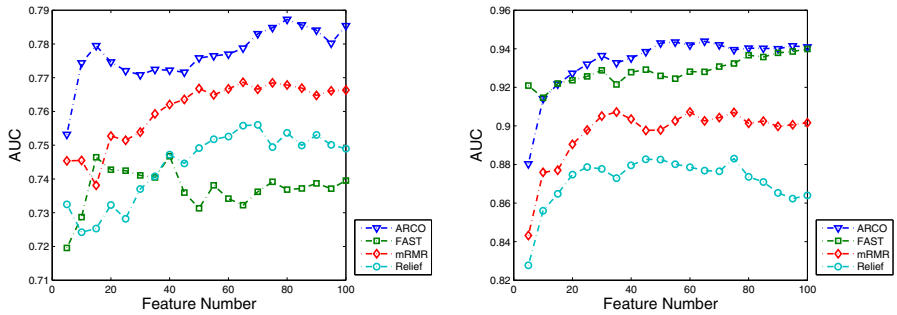


(a) Colon-NB (b) Leukemia-NB

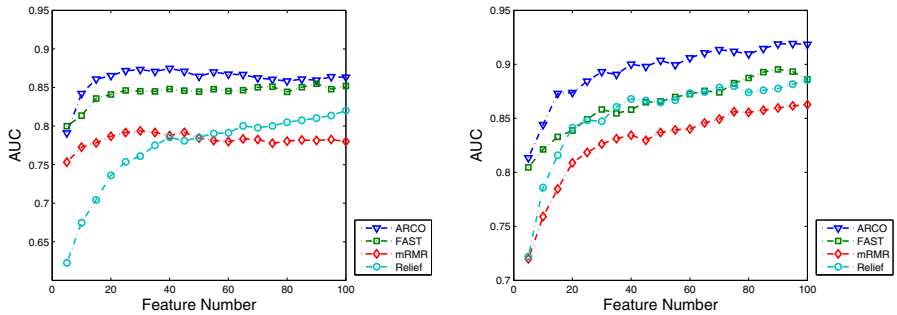


(c) CNS-NB (d) Lymph-NB

Figure 1. AUC Comparing among four feature selection methods using NB classifier



(a) Colon-1NN (b) Leukemia-1NN



(c) CNS-1NN (d) Lymph-1NN

Figure 2. AUC Comparing among four feature selection methods using 1NN classifier

Table 2. Results of the Wilcoxon signed-rank test between ARCO and the compared methods

	NB			1NN		
vs	FATS	mRMR	Relief	FATS	mRMR	Relief
Colon	20*w*-0*d*-0*l*	19*w*-1*d*-0*l*	4*w*-16*d*-0*l*	20*w*-0*d*-0*l*	3*w*-17*d*-0*l*	18*w*-2*d*-0*l*
Leukemia	8*w*-11*d*-1*l*	20*w*-0*d*-0*l*	20*w*-0*d*-0*l*	3*w*-17*d*-0*l*	20*w*-0*d*-0*l*	20*w*-0*d*-0*l*
CNS	18*w*-2*d*-0*l*	20*w*-0*d*-0*l*	20*w*-0*d*-0*l*	8*w*-12*d*-0*l*	20*w*-0*d*-0*l*	20*w*-0*d*-0*l*
Lymph	19*w*-1*d*-0*l*	12*w*-8*d*-0*l*	2*w*-18*d*-0*l*	16*w*-4*d*-0*l*	20*w*-0*d*-0*l*	20*w*-0*d*-0*l*

sification algorithms in this work. It would be interesting to combine ARCO and those algorithms specifically designed for AUC maximization. Hopefully, synergy between them will further boost the AUC of the final model.

6. Acknowledgments

This work is partially supported by two National Natural Science Foundation of China grants (No. 60802036 and No. U0835002).

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

[2] U. Brefeld and T. Scheffer. AUC maximizing support vector learning. In *Proceedings of the 22th International Conference on Machine Learning Workshop on ROC Analysis in Machine Learning*, Bonn, Germany, 2005.

[3] X. Chen and M. Wasikowski. FAST: a ROC-based feature selection metric for small samples and imbalanced data classification problems. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 124–132, New York, NY, USA, 2008.

[4] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the IEEE Computer Society Conference on Bioinformatics*, pages 523–528, Washington, DC, USA, 2003.

[5] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[6] C. Ferri, P. Flach, and J. Hernndez-Orallo. Learning decision trees using the area under the ROC curve. In *Proceedings of the 19th International Conference on Machine Learning*, pages 139–146, Sydney, Australia, 2002.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[10] D. J. Hand and R. J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[11] J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[12] E. L. Lehmann and H. J. M. D'Abrera. *Nonparametrics. Statistical Methods Based on Ranks*. McGraw Hill International Book Company, 1975.

[13] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[14] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 2002.

[15] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[16] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8:68–74, 2002.

[17] K. Tang, P. Suganthan, and X. Yao. Gene selection algorithms for microarray data based on least squares support vector machine. *BMC Bioinformatics*, 7:95, 2006.

[18] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)*. Morgan Kaufmann, 2005.