A Comparative Evaluation of Feature Ranking Methods for High Dimensional Bioinformatics Data

Jason Van Hulse, Taghi M. Khoshgoftaar, Amri Napolitano
jvanhulse@gmail.com, taghi@cse.fau.edu, amrifau@gmail.com
Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, Florida, USA

Abstract

Feature selection is an important component of data mining analysis with high dimensional data. Reducing the number of features in the dataset can have numerous positive implications, such as eliminating redundant or irrelevant features, decreasing development time and improving the performance of classification models. In this work, four filter-based feature selection techniques are compared using a wide variety of bioinformatics datasets. The first three filters, χ^2, Relief-F and Information Gain, are widely used techniques that are well known to many researchers and practitioners. The fourth filter, recently proposed by our research group and denoted TBFS-AUC (i.e., Threshold-Based Feature Selection technique with the AUC metric), is compared to these three commonly-used techniques using three different classification performance metrics. The empirical results demonstrate the strong performance of our technique.

Keywords: Feature selection; Bioinformatics; Threshold-based Feature Selection

1 Introduction

Feature selection [8] is an important step in the data mining and knowledge discovery process, particulary when the data has very high dimensionality. The objective of feature selection is to reduce the number of attributes in the dataset such that the selected features incorporate as much information from the entire dataset as possible. In the context of classification problems, this objective can be simply stated as the process of reducing the number of independent variables in order to optimize the performance of the learner on test data. Using too many features can hurt classification performance, particularly when many features are noisy or irrelevant. On the other hand, mistakenly eliminating important predictors can also decrease performance.

Generally speaking, feature selection techniques are often classified in two categories: *filter-based* and *wrapper-based*. Filter-based techiques select a feature subset without involving any learner. Wrapper algorithms utilize feedback from a classification algorithm to determine which features should be selected. Another way to categorize feature selection algorithms is as either *feature ranking* or *feature subset selection*. Feature ranking algorithms literally rank the features from most to least important; a user can then determine what the appropriate inclusion strategy is. For example, the user may select the 25 most significant features, and discard the remaining ones. Feature subset selection, on the other hand, selects a subset of features that perform well together, without necessarily determining which individual attributes are more significant than the others. This work considers only filter-based feature ranking techniques, which are the most common type studied in related work.

The bioinformatics application domain is one in particular where datasets often have a very large number of features. Compounding the problem, these datasets often have relatively few examples. Therefore, attribute selection is critical when building classification models for bioinformatics applications [27]. The objective of this work is to compare four filter-based ranking techniques for bioinformatics applications. The filter-based ranking techniques considered are χ^2, information gain (IG), ReliefF (RF) and TBFS-AUC. The first three techniques are commonly used, while the fourth, TBFS-AUC or simply AUC, is proposed by our research group.TBFS is an abbreviation for the threshold-based feature selection technique, and AUC is the version of this method that is evaluated in this work. The empirical case study blends the classification results from 17 bioinformatics datasets. Combining the results from a multitude of datasets improves the reliability of our work. In the experiments, classification models are built using the Naive Bayes learner.

The remainder of this paper is organized as follows. Section 2 discusses related work, while the feature selection algorithms are described in Section 3. The datasets used in the

IEEE IRI 2011, August 3-5, 2011, Las Vegas, Nevada, USA
978-1-4577-0966-1/11/$26.00 ©2011 IEEE

empirical study are discussed in Section 4, and the results of our experiments are presented in Section 5. Conclusions and directions for future work are presented in Section 6.

2 Related Work

Feature selection has received a significant amount of attention both in the bioinformatics domain and in data mining in general. Liu and Yu [18] provided a comprehensive survey of feature selection algorithms and presented an integrated approach to intelligent feature selection. Forman [6] investigated multiple filter-based feature ranking techniques for text categorization. Hall and Holmes [9] provide a benchmark comparison of several attribute selection methods. Molina et al. [20] evaluates feature selection methods using a variety of simulated datasets.

In the context of bioinformatics data, numerous biological data sources are amenable to data mining analysis, for example, protein sequences [26] and properties [16]; genetic codes [28]; mass spectroscopy results [17]. Much of the research on microarray analysis, where datasets often have thousands of features, has focused on improving classification models. Some researchers only use the standard array of feature ranking and subset evaluation filters and wrappers, coupled with traditional data mining techniques; these either analyze filters alone [7] or compare filters and wrappers [11] or filters, wrappers, and principal component analysis [19]. Others employ genetic algorithms [12] or minimum redundancy [23] to find the optimal subset of genes for classification purposes. Novel feature selection techniques have also been designed for microarray analysis [21, 3, 4]. Jong et al. [13] introduced methods for feature selection based on support vector machines (SVM). Ilczuk et al. [10] investigated the importance of attribute selection in judging the qualification of patients for cardiac pacemaker implantation.

3 Feature Selection Techniques

The three common filter-based feature ranking techniques considered in this work are chi-squared [29], information gain [9, 25, 29] and ReliefF [15]. All of these feature selection methods are available within Weka [29]. The chi-squared method (χ^2) utilizes the χ^2 statistic to measure the strength of the relationship between each independent variable and the class. Information Gain (IG) determines the significance of a feature based on the amount by which the entropy of the class decreases when considering that feature. Both χ^2 and IG utilize the method of Fayyad and Irani [5] to discretize continuous attributes. χ^2 and IG are bivariate techniques, considering the relationship between each attribute and the class, excluding the other independent variables.

Algorithm 1: Threshold-Based Feature Selection Algorithm

```
input :
    a. Dataset $D$ with features $X^j, j = 1, \dots, m$;
    b. Each instance $x \in D$ is assigned to one of two classes $c(x) \in \{P, N\}$;
    c. $|P| = |\{x \in D | c(x) = P\}|, |N| = |\{x \in D | c(x) = N\}|$;
    d. The value of attribute $X^j$ for instance $x$ is denoted $X^j(x)$;
    e. Metric $\omega$ = AUC.
output: Ranking $\mathcal{R} = \{r^1, r^2, \dots, r^m\}$ where attribute $X^j$ is the $r^j$-th
        most significant attribute as determined by metric $\omega$.
for $X^j, j = 1, \dots, m$ do
    Normalize $X^j \mapsto \hat{X}^j = \frac{X^j - \min(X^j)}{\max(X^j) - \min(X^j)}$, $\hat{X}^j \in [0, 1]$;
    for $t \in [0, 1]$ do
        Compute Basic Metrics:
        Classification Rule 1:
        $\forall\ x \in D, \hat{c}^t(x) = P \iff \hat{X}^j(x) > t$, otherwise
        $\hat{c}^t(x) = N$.
        $TP(t) = \left|\{x | (\hat{c}^t(x) = P) \cap (c(x) = P)\}\right|$,
        $TN(t) = \left|\{x | (\hat{c}^t(x) = N) \cap (c(x) = N)\}\right|$,
        $FP(t) = \left|\{x | (\hat{c}^t(x) = P) \cap (c(x) = N)\}\right|$,
        $FN(t) = \left|\{x | (\hat{c}^t(x) = N) \cap (c(x) = P)\}\right|$,
        $TPR(t) = \frac{|TP(t)|}{|P|}$, $TNR(t) = \frac{|TN(t)|}{|N|}$,
        $FPR(t) = 1 - TNR(t)$
    Compute Final Metric:
    $\omega^1(\hat{X}^j)$ = Area under the curve generated by
    $(FPR(t), TPR(t)), t \in [0, 1]$
    Compute the same basic metrics and final metric (denoted as $\omega^2$) as
    listed above, but using:
    Classification Rule 2: $\forall\ x \in D, \hat{c}^t(x) = N \iff \hat{X}^j(x) > t$,
    otherwise $\hat{c}^t(x) = P$.
    $\omega(\hat{X}^j) = \max\left(\omega^1(\hat{x}^j), \omega^2(\hat{x}^j)\right)$
Create attribute ranking $\mathcal{R}$ using $\omega(\hat{X}^j) \forall j$
```

Relief [15, 14] randomly samples an example from the data and finds its nearest neighbor from the same and opposite class. The values of the attributes of the nearest neighbors are compared to the sampled instance and used to update relevance scores for each attribute. This process is repeated for m examples, as specified by the user. ReliefF (RF) extends Relief by handling noise and multiclass data sets [15]. RF is implemented within Weka [29] with the "weight nearest neighbors by their distance" parameter set to false.

3.1 Threshold-Based Feature Selection Technique

This section describes the TBFS method for feature ranking. Similar to the χ^2 and IG, TBFS is a bivariate procedure; each attribute is evaluated against the class, independent of all other features in the dataset. After normalizing each attribute to have a range between 0 and 1, simple classifiers are built for each threshold value $t \in [0, 1]$ according to two different classification rules. For classification rule 1, examples with a normalized value greater than

Dataset Name	Abbreviation	# Attributes	# Total	# Positive	% Positive
ECML Pancreas	ECML	27680	90	8	8.9%
Central Nervous System	CNS	7130	60	21	35.0%
Colon	Colon	2001	62	22	35.5%
DLBCL Tumor	Tum	7130	77	19	24.7%
Lymphoma	Lymph	4027	96	23	24.0%
DLBCL	DLB	4027	47	23	48.9%
Lung Cancer	LC	12534	181	31	17.1%
Acute Lymphoblastic Leukemia	ALL	12559	327	79	24.2%
Prostate	Pros	12601	136	59	43.4%
Mll Leukemia	MLL	12583	72	20	27.8%
Breast Cancer	Brst	24482	97	46	47.4%
All Aml Leukemia	AAL	7130	72	25	34.7%
Translation Initiation	Tran	925	13375	3312	24.8%
Ovarian Cancer	Ov	15155	253	91	36.0%
DLBCL NIH	NIH	7399	240	103	42.9%
Lung	Lung	12601	203	65	32.0%
Brain Tumor	Brain	27679	90	23	25.6%

Table 1. Bioinformatics Datasets

Filter	ECML	CNS	Colon	Tum	Lymph	DLB	LC	ALL	Pros	MLL	Brst	AAL	Tran	Ov	NIH	Lung	Brain
χ^2	.733	.622	.868	.924	.904	.955	.998	.988	.707	.961	.633	.980	.900	.994	.586	.966	.916
IG	.775	.628	.866	.929	.867	.955	.991	.990	.687	.952	.622	.980	.902	.993	.580	.964	.936
RF	.887	.567	.843	.944	.858	.969	.992	.987	.673	.929	.749	.962	.752	.988	.569	.931	.768
AUC	.912	.589	.863	.942	.893	.967	.998	.994	.734	.936	.636	.979	.906	.995	.573	.976	.927
Average	.827	.602	.860	.935	.881	.961	.995	.990	.700	.944	.660	.976	.865	.992	.577	.959	.887
St Dev	.087	.029	.011	.010	.022	.008	.004	.003	.026	.015	.060	.009	.075	.003	.008	.020	.080

Table 2. Filter Results by Dataset, AROC Performance Metric

t are classified P while examples with a normalized value less than t are classified as N (assuming each instance x is assigned to one of two classes $c(x) \in \{P, N\}$). For classification rule 2, examples with a normalized value greater than t are classified N while examples with a normalized value less than t are classified as P. Two different classification rules must be considered to account for the fact that for some attributes, large values of the attribute may have a greater correlation with the positive class, while for other attributes, large values of the attribute may have a greater correlation with the negative class. The AUC is calculated as the area under the curve generated from the true positive and false positive rates (the receiver operating characteristic or ROC curve). Finally, the metric resulting from the classification rule which provides the largest value is used as the relevancy measure for that attribute relative to the AUC.

The AUC is primarily used to measure the performance of classification models, using the posterior probabilities computed by such models to classify examples as either negative or positive depending on the classification threshold. The normalized attribute values can be thought of as posterior probabilities, e.g., $p(P \mid x) = \hat{X}^j(x)$ for classification rule 1, and the AUC is computed against this "posterior." Intuitively, attributes where positive and negative examples are evenly distributed along the distribution of X produce weak measures and poor relevancy scores in a similar manner that poor predictive models have positive and negative examples evenly distributed along the distribution of the posterior probability produced by the model.

TBFS is a substantial extension of the FAST algorithm [1]. FAST is based on the area under a ROC curve generated by moving the decision boundary of a single feature classifier with thresholds placed using an even-bin distribution. FAST calculates a ROC curve by discretizing the distribution, while TBFS-AUC does not require discretization, making it more precise and eliminating the often vexing question of how wide the bins should be. TBFS can easily be extended to include additional metrics.

4 Datasets

The datasets utilized in our experiments are listed in Table 1. All of the datasets come from the bioinformatics application domain, and all but two (Translation and Ovarian) are microarray expression datasets. Table 1 provides the number of attributes, number of total examples, number of positive examples and the percentage of positive examples for each dataset. Note that all of the datasets used in this work have a binary dependent variable. Further note that these datasets exhibit a wide distribution of class skew (i.e., the percentage of positive examples).

For the Ovarian cancer dataset [24], the researchers took serum samples from patients with and without cancer and ran them through a mass spectroscopy machine, giving them 15155 separate mass/charge values. That is, all the proteins in the sample were ionized and deflected through

Filter	ECML	CNS	Colon	Tum	Lymph	DLB	LC	ALL	Pros	MLL	Brst	AAL	Tran	Ov	NIH	Lung	Brain
χ^2	.313	.508	.792	.788	.790	.960	.984	.935	.687	.880	.632	.937	.726	.992	.529	.917	.768
IG	.332	.518	.792	.799	.694	.961	.970	.942	.662	.891	.603	.937	.727	.991	.521	.922	.794
RF	.525	.444	.788	.836	.699	.972	.931	.928	.630	.836	.735	.915	.472	.982	.516	.853	.548
AUC	.624	.496	.781	.827	.764	.974	.981	.973	.686	.858	.632	.935	.729	.993	.527	.959	.792
Average	.449	.492	.788	.813	.736	.967	.967	.944	.666	.866	.650	.931	.663	.989	.523	.913	.725
St Dev	.151	.033	.005	.023	.048	.007	.024	.020	.027	.024	.058	.011	.127	.005	.006	.044	.119

Table 3. Filter Results by Dataset, APRC Performance Metric

a magnetic field such that proteins with a different ratio of mass to charge would behave differently and thus be detected separately. Thus, the different mass/charge values reflect the relative abundance of different proteins in each serum sample. The Translation dataset [22] is based on a set of mRNA sequences for different genes found in vertebrates and plants, each of which is annotated with its translation initiation point (the ATG which represents the start of the gene). To generate features for a given instance, the upstream and downstream parts of the sequence (that is, the parts before and after the translation initiation point) are searched for all 20 amino acids and the stop codon, as well as for all two-amino-acid sequences (and for the pairs which include the stop codon). The number of times each amino acid or pair of amino acids is found is the value for that feature.

5 Results

The results of experiments using four filter-based feature selection techniques are presented in this section. The base learner used in these experiments was Naive Bayes (NB). Experiments were conducted in the Weka [29] data mining suite. The TBFS method was implemented by our research group within the Weka framework. Ten-fold cross validation was used to evaluate model performance. Each of the 17 datasets listed in Table 1 was partitioned into ten equal-sized subsets. Nine of the subsets were combined to form the training dataset, and the remaining partition was used as the test data. Feature selection is performed with one of the filters using the training data, and a Naive Bayes model is constructed using the training dataset with the reduced features. For each filter, the 10 most significant attributes were utilized. Other values for the number of selected features can also be used. It was not the intention to optimize this parameter, or to evaluate its impact on the filters. Instead, we selected a reasonable value which showed differentiation between the techniques. When the number of selected features increases, the performance of the filters converges as all of the filters select the most important features. Since the goal of feature selection is to find a small subset of attributes, it makes sense to compare the filters using a small value for this parameter. Other values should be evaluated in future work. The model is then evaluated using the test data (hold-out partition). This procedure is repeated such that each partition in the cross validation process is used as a test dataset once. In addition, cross-validation is repeated four times to include additional randomization in the evaluation process.

Models are evaluated using three performance metrics: the area under the receiver operating characteristics (ROC) curve (AROC); the area under the precision-recall (PR) curve (APRC); and the F-measure.

For each example x in the test dataset, the NB algorithm outputs posterior probabilities, i.e., $p(P \mid x)$ and $p(N \mid x)$, where the class c of x is either P or N (this work only considers binary classification problems). The predicted class of x, denoted $\hat{c}(x)$, is determined based on a threshold $t \in [0, 1]$. If $p(P \mid x) > t$, then $\hat{c}(x) = P$ given threshold t; otherwise, $\hat{c}(x) = N$. The ROC curve is obtained by computing the false positive and true positive rates at each possible threshold. The false positive rate at t is the number of examples falsely classified as P, and the true positive rate at t is the number of correctly classified P examples. The AROC is computed as the area under the ROC curve, and is a value between 0 and 1. The PR curve is determined by plotting the recall and precision as the threshold is varied from 0 to 1. Recall is equivalent to the true positive rate, while precision is the percentage of examples that are predicted to be positive that are actually from the positive class. APRC is computed as the area under the PR curve. For highly skewed datasets (where the ratio of positive to negative examples is very low), APRC may be a more appropriate metric than AROC [2]. Similar to AROC, the range of APRC is between 0 and 1, with a better classifier closer to 1. The final metric, the F-measure, is the harmonic mean of the recall and precision.

Table 2 compares the four filters using the AROC performance metric, while Tables 3 and 4 show the results for APRC and F-measure, respectively. For each filter and dataset, the average value over the 4 runs of cross validation is presented. The average value for a given dataset, and the standard deviation of the 4 values are also provided in the last 2 rows of each table. For some datasets, all four filters perform similarly. For example, on dataset LC, the values of AROC range from .998 to .991. Similarly, with dataset NIH, all four filters perform equally poorly, with the AROC ranging from .569 to .586. It is most interesting to consider the datasets where there is a substantial difference between filters, such as ECML, where AUC has an AROC of .912,

Filter	ECML	CNS	Colon	Tum	Lymph	DLB	LC	ALL	Pros	MLL	Brst	AAL	Tran	Ov	NIH	Lung	Brain
χ^2	.367	.519	.721	.798	.706	.930	.976	.953	.513	.827	.406	.922	.668	.947	.501	.862	.787
IG	.305	.508	.723	.775	.645	.930	.969	.944	.510	.839	.302	.922	.672	.949	.501	.864	.802
RF	.596	.360	.743	.766	.612	.931	.940	.945	.655	.817	.678	.901	.434	.951	.500	.799	.555
AUC	.631	.518	.725	.812	.648	.922	.968	.955	.633	.772	.282	.922	.682	.956	.490	.873	.788
Average	.475	.476	.728	.788	.653	.928	.963	.949	.578	.814	.417	.916	.614	.951	.498	.850	.733
St Dev	.163	.078	.010	.021	.039	.004	.016	.005	.077	.029	.182	.010	.120	.004	.005	.034	.119

Table 4. Filter Results by Dataset, F-measure Performance Metric

	AROC						APRC						F-measure				
Filter	Simple	Easy	Mod	Hard	Mean	St Dev	Simple	Easy	Mod	Hard	Mean	St Dev	Easy	Mod	Hard	Mean	St Dev
χ^2	.980	.943	.864	.637	.861	.144	.954	.834	.732	.450	.773	.191	.946	.799	.526	.730	.202
IG	.979	.941	.869	.629	.860	.144	.954	.845	.712	.457	.768	.191	.942	.801	.492	.715	.222
RF	.971	.937	.822	.640	.845	.142	.930	.836	.645	.495	.742	.186	.934	.736	.548	.717	.188
AUC	.985	.939	.900	.633	.872	.145	.969	.843	.731	.549	.796	.163	.945	.794	.555	.740	.193
Range	$> .95$	.9-.95	.8-.9	$< .8$			$> .9$	.8-.9	.6-.8	$< .6$			$> .9$	.7-.9	$< .7$		

Table 5. Summary of Performance

compared to .733 for χ^2. A similar pattern for the ECML dataset is evident relative to APRC and F-measure, where AUC is substantially better than the other filters.

Table 5 summarizes the performance of the filters for all of the performance metrics. First, for each metric, the datasets were categorized into one of three (for F-measure) or four (for AROC and APRC) categories, based on the average value over all four filters. The criteria used for this categorization is provided in the last row of Table 5. For example, datasets were categorized as "simple" to learn if the average AROC was greater than .95. There were six datasets (DLB, LC, ALL, AAL, Ov and Lung), and the average AROC for χ^2 for these datasets is .980. The average and standard deviation for each metric over all 17 datasets is also provided in Table 5.

The TBFS filter with AUC obtains the highest average AROC, APRC and F-measure. Particularly for the harder datasets, AUC performed very well relative to the other filters. Not only does AUC perform well on average, but it is also a relatively stable technique. For example, consider Table 2. AUC had the best performance on a dataset six times, and was never the worst of the four filters for any dataset. Similarly, relative to APRC, AUC is the best technique for six of the datasets, and is the worst technique for only one dataset (Colon). χ^2 is also a strong filter, with the second-highest average AROC, APRC and F-measure metrics in Table 5. Relative to the AROC, χ^2 also obtains the best performance for six datasets, but unlike the AUC filter, obtains the worst performance for two datasets (Tum and ECML).

The performances of RF and IG are clearly inferior to that of χ^2 and AUC, with RF generally performing the worst in our experiments. RF had the lowest average AROC and APRC, and was the worst-performing filter for 12 of the 17 datasets (AROC) and 11 of the 17 datasets (APRC).

It is interesting to note the differences between the different performance metrics AROC, APRC and F-measure. In large part, the conclusions derived from these three metrics agree with one another. Indeed, the correlation coefficient between average AROC and APRC values for the AUC filter is 0.883, so there is clearly a high degree of similarity. However, there are some interesting differences to note. For example, the filters exhibit a much larger degree of instability with the F-measure. On the Brst dataset, the best technique (RF) obtains a .678 F-measure, while AUC obtains a .282 F-measure. Relative to AROC and APRC, the spread between best and worst filter was much smaller. One possible reason for this is that the F-measure uses a fixed threshold $t = 0.5$ when determining the predicted class of an example. AROC and APRC, on the other hand, aggregate performance across all thresholds. In many situations, this fixed decision threshold may not be appropriate.

6 Conclusions

Data analysis conducted on high-dimensional bioinformatics data can often benefit from the use of feature selection techniques. This work presents an empirical comparison of four filter-based feature ranking techniques, χ^2, IG, RF and AUC, the last of which was recently proposed by our research group. Seventeen datasets from the bioinformatics domain were considered, and Naive Bayes models were built in the Weka data mining tool and evaluated using three performance metrics. The results from all of these experiments were combined in order to reach some conclusion regarding the relative effectiveness of these techniques.

The TBFS filter with the AUC metric performed very well in our experiments. Relative to all three metrics, this filter outperformed the other three techniques. χ^2 also performed well, and both χ^2 and AUC were clearly superior to the other methods, IG and RF. Future work can include additional feature ranking techniques, base learners besides Naive Bayes, performance metrics and datasets/application domains.

References

[1] 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*, pages 124–132, New York, NY, USA, 2008. ACM.

[2] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 115–123, 1210 West Dayton Street, Madison, WI, 53706 USA, 2006. University of Wisconsin-Madison.

[3] D. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse. Comparative analysis of DNA microarray data through the use of feature selection techniques. In *Proceedings of the IEEE International Conference on Machine Learning and Applications*, pages 147–152, Washington, D.C., December 12-14, 2010.

[4] D. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse. Feature selection algorithms for mining high-dimensional DNA microarray data. *Data Intensive Computing*, 2011. Editors: Borko Furht and Armando Escalante. In Press.

[5] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[6] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

[7] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

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

[9] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):392–398, November/December 2003.

[10] G. Ilczuk, R. Mlynarski, W. Kargul, and A. Wakulicz-Deja. New feature selection methods for qualification of the patients for cardiac pacemaker implantation. *Computers in Cardiology*, 34(2-3):423–426, 2007.

[11] I. Inza, P. Larraaga, R. Blanco, and A. J. Cerrolaza. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 31(2):91 – 103, 2004. Data Mining in Genomics and Proteomics.

[12] T. Jirapech-Umpai and S. Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(1):148, 2005.

[13] K. Jong, E. Marchiori, M. Sebag, and A. van der Vaart. Feature selection in proteomic pattern data with support vector machines. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Oct 7-8 2004.

[14] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new solution. In *AAAI '92: Proc. 10th Nat'l Conf. on Artificial Intelligence*, number 10, pages 129–134. John Wiley & Sons, Ltd., July 1992.

[15] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182. Springer Verlag, 1994.

[16] B. J. Lee, H. G. Lee, J. Y. Lee, and K. H. Ryu. Classification of enzyme function from protein sequence based on feature representation. pages 741–747, Oct. 2007.

[17] L. Li, H. Tang, Z. Wu, J. Gong, M. Gruidl, J. Zou, M. Tockman, and R. A. Clark. Data mining techniques for cancer detection using serum proteomic profiling. *Artificial Intelligence in Medicine*, 32(2):71 – 83, 2004.

[18] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[19] F. Model. Feature selection for dna methylation based cancer classification. *Bioinformatics*, 17:157–164(8), June 2001.

[20] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 306–313, Washington, DC, USA, 2002. IEEE Computer Society.

[21] S. D. Peddada, E. K. Lobenhofer, L. Li, C. A. Afshari, C. R. Weinberg, and D. M. Umbach. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, 19(7):834–841, 2003.

[22] A. G. Pedersen and H. Nielsen. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 226–233. AAAI Press, 1997.

[23] 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.

[24] E. F. Petricoin III, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359(9306):572–577, 2002.

[25] J. R. Quinlan. *C4.5: Programs For Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[26] P. Radivojac, N. V. Chawla, A. K. Dunker, and Z. Obradovic. Classification and knowledge discovery in protein databases. *Journal of Biomedical Informatics*, 37(4):224–239, 2004. Biomedical Machine Learning.

[27] Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[28] Y. Sun, M. Robinson, R. Adams, R. te Boekhorst, A. Rust, and N. Davey. Using feature selection filtering methods for binding site predictions. volume 1, pages 566–571, July 2006.

[29] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition, 2005.