Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference

Optimizing Wrapper-Based Feature Selection for Use on Bioinformatics Data

Randall Wald, Taghi M. Khoshgoftaar, Amri Napolitano
Florida Atlantic University
Email: {rwald1, khoshgof}@fau.edu, amrifau@gmail.com

Abstract

High dimensionality (having a large number of independent attributes) is a major problem for bioinformatics datasets such as gene microarray datasets. Feature selection algorithms are necessary to remove the irrelevant (not useful) and redundant (contain duplicate information) features. One approach to handle this problem is wrapper-based subset evaluation, which builds classification models on different feature subsets to discover which performs best. Although the computational complexity of this technique has led to it being rarely used for bioinformatics, its ability to find the features which give the best model make it important in this domain. However, when using wrapper-based feature selection, it is not obvious whether the learner used within the wrapper should match the learner used for building the final classification model. Furthermore, this question may depend on other properties of the dataset, such as difficulty of learning (general performance without feature selection) and dataset balance (ratio of minority and majority instances). To study this, we use nine datasets with varying levels of difficulty and balance. We find that across all datasets, the best strategy is to use one learner (Naïve Bayes) inside the wrapper regardless of the learner which will be used outside. However, when broken down by difficulty and balance levels, our results show that the more balanced and less difficult datasets work best when the learners inside and outside the wrapper match. Thus, the answer to this question will depend on properties of the dataset.

1 Introduction

The rise of advanced data-gathering techniques in the field of bioinformatics has led to the paradox of Big Data: there is so much data available, none of it can directly understood by researchers and practitioners. Fortunately, computing capabilities and the fields of data mining and machine learning have also grown to keep up with this increase in data. One particular problem often found in bioinformatics datasets is high dimensionality: having a very large number of independent features, or attributes. Gene microarray datasets are a perfect example of this problem: for each tissue sample, the gene microarray will record the gene expression levels for thousands (or even tens of thousands) of gene probes, but in practice only a small handful of these are actually relevant to the underlying biological question (e.g., finding which genes distinguish cancerous tissues from non-cancerous tissues, or creating genetic signatures to predict patient response to cancer treatment). Thus, to make sense of these large datasets, feature selection techniques are needed in order to eliminate features which are irrelevant (not containing information pertinent to the biological question) or redundant (containing information already found in other features).

Three forms of feature selection are commonly used in the literature: filter-based feature ranking, filter-based subset evaluation, and wrapper-based subset evaluation. The first of these uses a statistical filter to assign a score to each feature, and then ranks all features based on these scores. The second also uses statistical filters, but chooses techniques that apply to whole feature subsets (rather than individual features. Thus, filter-based subset evaluation can detect redundant features — but it pays for this ability by being much more computationally expensive than filter-based feature ranking. Wrapper-based subset evaluation also considers whole subsets, but scores each based on the performance of a classification model built with those features, rather than using a statistical filter. This enables wrappers to find the features which actually are best for building classification models,– rather than just optimizing some arbitrary statistical filter — but typically such models are more computationally expensive than even filter-based subset evaluation. This expense has led to wrappers being neglected in the bioinformatics literature, despite their potential to discover the most important gene subsets.

In addition, even when considering other application domains, most works employing wrappers use the same learner both for selecting features and for building the final classification model. While this makes intuitive sense (as this approach should give the features that work best with that learner), it is important to test hypothesis and understand how they apply to real-world data. Thus, in our study we consider the effects of matching different choices of "wrapper" (or "internal") learners with different choices of "classification" (or "external") learners, to discover whether or not the optimal strategy is to always make these the same. In addition, we seek to find whether the choice of optimal strat-

egy may depend on other properties of the dataset, such as its overall difficulty (i.e., "difficulty of learning," how challenging the dataset is to learn from in general) and its balance level (i.e., the fraction of instances found in the minority class of a two-class dataset). It is possible that previous research on wrappers found one strategy (vis-a-vis matching the internal and external learners) to be optimal solely because these studied one type of dataset, and that other strategies dominate on other dataset types. To understand these influences, we consider three levels of difficulty and three levels of balance, with nine datasets spread across this three-by-three matrix. Three learners (5-Nearest Neighbor, Logistic Regression, and Naïve Bayes) are used in our study.

Our experiments show that these two properties do lead to different choices of optimal strategy. In particular, although overall we find the one choice of learner (Naïve Bayes) is always the best choice of wrapper learner, this statement is only true when considering the most imbalanced datasets on their own. None of the levels of difficulty show this pattern when isolated from the rest: the datasets with Easy and Moderate difficulty show Naïve Bayes to be the best wrapper learner for two of the three choices of classification learner, with the third (5-Nearest Neighbor for Easy datasets, Logistic Regression for Moderate datasets) working best when paired with itself inside the wrapper. Considering the other two levels of balance (balanced and slightly imbalanced), the balanced datasets always showed the best performance when the wrapper learner matched the classification learner, but the slightly imbalanced datasets showed the best performance only when the wrapper learner did *not* match the wrapper learner. Overall, these results show that whether or not matching the wrapper learner with the classification learner is the optimal strategy will depend heavily on the properties of the datasets, and that in general, the more challenging the dataset (in terms of difficulty of learning or balance), the more likely it is that matching the wrapper learner and the classification learner will not be an optimal strategy.

The structure of this paper is as follows: Section 2 contains related work on the topic of subset evaluation, especially as it pertains to bioinformatics. Section 3 reviews our methods for wrapper feature selection, classification, and performance evaluation. Section 4 holds the details of our datasets and case study. Section 5 presents our results and discussion of these results. Finally, in Section 6 we make our concluding remarks and suggestions for future research.

2 Related Work

Due to its greater computational complexity, wrapper-based feature selection has rarely been used in the context of bioinformatics. Although some other works (Kohavi and John 1997; Peng, Long, and Ding 2005; Zhu, Ong, and Dash 2007) have considered wrapper-based feature selection in a broader context, including a handful of bioinformatics datasets along with datasets from other application domains, relatively few have specifically focused on its use in this domain. In their broad review of feature selection for bioinformatics, Saeys et al. (Saeys, Inza, and Larranaga 2007) discuss wrapper feature selection, and note that to alleviate the computational complexity of building models based on all possible gene subsets, most prior works have used some form of randomized subset search such as a genetic algorithm, simulated annealing, or randomized hill climbing. However, they do note that some work has considered a more systematic approach to wrapper selection. Inza et al. (Inza et al. 2004) compare filter-based feature ranking with wrapper-based subset selection, using two bioinformatics datasets and six feature ranking techniques (two for continuous data, four for discrete) along with four choices of learner. They find that wrapper feature selection gives better performance than filter-based ranking, but at a high computational cost. Xiong et al. (Xiong, Fang, and Zhao 2001) also consider wrapper feature selection, using three learners and three datasets. They found that selecting more than one top feature was able to improve performance over using just one feature, and that a more advanced search technique capable of backtracking showed greater performance than simple forward selection. Leung and Hung (Leung and Hung 2008) proposed an ensemble hybrid approach using both filters and wrappers, and compare it with a non-ensemble hybrid approach using six bioinformatics datasets, showing that their proposed technique performs better. Wang et al. (Wang et al. 2005) compare all three forms of feature selection, using four filter-based rankers, one filter-based subset evaluator, and three classifiers for both wrapper selection and final classification. Results are evaluated using two gene microarray datasets, and the authors find that on the first dataset, all three techniques are very consistent in terms of one gene found to have extremely high connection to the class in question; more varied results are found on the second dataset. Nonetheless, they find that the filter- and wrapper-based subset selection approaches can give good performance while selecting a smaller subset of features.

Overall, although some research has considered the use of wrapper-based feature selection on bioinformatics datasets, none have considered whether the learner used inside the wrapper should match the learner applied outside the wrapper to build the final classification model. In addition, while some previous works using wrapper-based feature selection for bioinformatics have considered imbalanced data (by using appropriate performance metrics to give meaningful measurement of the performance of classifiers on imbalanced data), none has considered a variety of datasets exhibiting different levels of class imbalance in order to discover properties which vary across such datasets. And no previous work has looked at how the general difficulty of learning from a given dataset might specifically affect the performance of wrapper-based feature selection, or how this difficulty might affect other aspects of wrapper-based feature selection (such as whether matching the learners inside and outside the wrapper gives optimal performance).

3 Methods

A total of three learners were used along with the wrapper feature selection technique in this case study. The wrapper technique itself is presented in Section 3.1, while the learners are presented individually in Section 3.2, and our performance metric and evaluation procedure are discussed in Section 3.3.

3.1 Wrapper Feature Selection

The basic premise of wrapper feature selection is building a model using a potential feature subset and using the performance of this model as a score for the merit of that subset (Kohavi and John 1997). As with any model-building process, a number of choices must be made in how to build and evaluate the model. First of all, while this model could be built using the full training set and then have its performance evaluated against that same training set, this would potentially lead to overfitting: building models which memorize the data rather than learning general properties of the data. Thus, for our experiments the wrapper process uses cross-validation (as discussed further in Section 3.3): the training set is divided into five parts, and models (using the potential feature subsets) are built on only four parts, and evaluated on the fifth. This process is repeated (by changing which folds are used for building the model) until all folds have been the evaluation fold exactly once, and the results are averaged to give the merit of the potential feature subset.

Also, when using wrapper feature selection, it is important to consider the performance metric used within the wrapper process. Just as imbalanced data can affect the performance of models used for final classification, it can give misleading results when used within wrapper feature selection. Thus, all models built within the wrapper feature selection framework used the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) metric (discussed further in Section 3.3).

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 final 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 current best-known set, or when 100 features have been selected (whichever happens first).

3.2 Learners

Three learners were chosen for our analysis: 5-Nearest Neighbor (5-NN), Logistic Regression (LR), and Naïve Bayes (NB). These were all chosen due to their relative ease of computation and their dissimilarity from one another. As wrapper feature selection necessarily involves building an extremely large number of models (using the Greedy Stepwise search technique, if k features are selected from a dataset containing n features, and $k \ll n$, approximately $k \times n$ models must be built), only simple models could be used. All models were built using the WEKA machine learning toolkit (Hall et al. 2009), using default parameters unless otherwise specified. Due to space limitations, we only give a brief outline of these techniques; for further information, we direct readers to Witten and Frank (Witten, Frank, and Hall 2011).

5-NN is a lazy instance-based learner which does not build a model per se but which uses the training data directly to make predictions about the test data. In particular, to classify a given instance, it finds the five nearest neighbors from the training set for that instance, and then has these vote (using weight by 1/distance) on the proper class value. LR is a simple regression model which uses the logistic function to normalize the probability between 0 and 1. NB is a Bayesean learner which uses Bayes's Theorem to find the posterior probability of the class values given the observed feature values. Although NB makes the naïve assumption that all feature values are statistically independent, it has been show to give good performance even when this assumption is not true (Lewis 1998).

3.3 Performance Measurement and Cross-Validation

While the choice of performance metric is important in any data mining study, this is especially important for considering wrapper-based feature selection, as the wrapper itself will use this performance metric to grade the subsets. The presence of imbalanced data also highlights the importance of this choice, in order to ensure that minority-class instances (positive instances) do not all end up misclassified. For this reason, we use Area Under the Receiver Operating Characteristic Curve (AUC) as our metric both inside the wrapper and for final classification. 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.

In addition to its use within the wrapper feature selection step, cross-validation was also used for building and testing the final classification models. In both cases, the cross-validation process begins by dividing the data into N equal-size subsets (folds), and then models are built (trained) on $N - 1$ of these and tested on the Nth fold, called the hold-out fold. This process is repeated N times, so that each fold is used as the hold-out fold exactly once. For the cross-validation within the wrapper procedure, we chose $N = 5$ and performed cross-validation once per feature subset being tested (as discussed in Section 3.1). However, we also used cross-validation for building the final classification models, and for this we used four runs of five-fold cross-validation ($N = 5$). This explains our decision to use only one run of five-fold cross-validation within the wrapper: due to our use of four runs of five-fold cross-validation for the external model-building process, we're already performing the wrapper step 20 times. Each run of the wrapper process already involves building a great many models, and thus increasing the number of models built within the wrapper (by switching to ten-fold cross-validation or by using more than one run) would significantly affect the computational load.

The AUC metric is calculated by collecting the results across all five folds. As noted, to further validate our results we performed the external cross-validation process a total of four times, and all results presented are the average across these four values.

Dataset Difficulty	Name	# Minority Instances	Total # of Instances	% Minority Instances	# of Attributes	Average AUC
Easy	**BCancer 50k**	200	400	50.00%	54614	0.85636
	Lung	64	203	31.53%	12601	0.86851
	ALL	79	327	24.16%	12559	0.84748
Moderate	**Prostate**	59	136	43.38%	12601	0.78225
	Colon	22	62	35.48%	2001	0.79413
	Brain Tumor	23	90	25.56%	27680	0.72096
Hard	**DLBCL NIH**	102	240	42.50%	7400	0.58527
	Chanrion 2008	52	155	33.55%	22657	0.67207
	Pawitan 2005	40	159	25.16%	12066	0.61082

Table 1: Details of the Datasets

4 Case Study

In this case study, we consider nine datasets across the domains of bioinformatics and patient response prediction. A summary of these datasets is found in Table 1. All datasets are gene microarray datasets. That is, the features represent the expression levels of various gene probes designed to target different parts of a cell's DNA sequence, and the class values (all of which are binary) come from whether a patient has cancer, what type of cancer a patient has, or whether the patient responded well to a particular cancer treatment. In particular, the Chanrion 2008 and Pawitan 2005 datasets come from the domain of patient response prediction, while all other datasets pertain to cancer detection or identification. Due to space limitations, we cannot individually discuss each dataset; further details for most datasets may be found in (Van Hulse, Khoshgoftaar, and Napolitano 2011), while information about the first, second-to-last, and last dataset are found in (Dittman et al. 2010), (Chanrion et al. 2008), and (Pawitan et al. 2005), respectively.

The last column, Average AUC, refers to the classification performance on these datasets when building models without feature selection. This is used to show that some of these datasets are notable for being difficult to model (such that models do not perform well), while others are particularly easy. One of the goals of our experiments was to determine how different dataset characteristics affect the optimal choices for wrapper feature selection, and this "difficulty of learning" value (also known as "dataset difficulty") was chosen as an important characteristic to study. The values in the table were calculated using a set of six different classification models: 5-NN, MLP, NB, SVM, and two versions of a C4.5 decision tree (C4.5 D and C4.5 N). Descriptions of the 5-NN and NB learners are found in Section 3.2. MLP is a multi-layer perceptron-based learner, with a single hidden layer contain 3 nodes and 10% of the data held back for validation of when to stop the backpropogation-based learning process. SVM is a Support Vector Machine designed to find the maximal margin hyperplan separating the classes, with a complexity constant of 5.0 and the `buildLogisticModels` parameter set to `true`. C4.5 D is the C4.5 decision tree classifier with the default parameter values. C4.5 N is the same classifier but with Laplace smoothing enabled and pruning disabled. All of these learners are available using the WEKA Data Mining toolkit (Witten, Frank, and Hall 2011), and all default values were used unless otherwise specified. Note that the results from these classifiers (without feature selection) were used only to determine the difficulty of the datasets and have no further bearing on the rest of the experiment.

As discussed, one major goal of our experiments was to discover the influence of dataset characteristics on the performance of different wrapper selection strategies. In particular, we chose to focus on two different characteristics: dataset difficulty and balance level. Dataset difficulty was defined based on the average AUC performance (as outlined above), and balance level was specified by considering the percentage of instances found in the minority class (as all of our datasets are binary, there is only one minority class). To facilitate our experiments, we created three levels of each of these factors. For difficulty of learning, the datasets were divided into Easy (Average AUC ≥ 0.8), Moderate (Average AUC < 0.8 and ≥ 0.7), and Hard (Average AUC < 0.7). For balance level, we divided the datasets into Balanced (% Minority $> 40\%$), Slightly Imbalanced (% Minority $\leq 40\%$ and $\geq 26\%$), and Imbalanced (% Minority $< 26\%$). In Table 1, we separated the Easy, Moderate, and Hard datasets into different groupings, and sort by balance level within each grouping; in addition, all Balanced datasets have their names printed in **bold**, and all Imbalanced datasets have their names printed in *italics*. Note that there is exactly one dataset for each combination of balance level and difficulty of learning.

5 Results

The results of our experiments are presented in Tables 2 through 4. Each table reflects the average AUC values found when selecting features through wrapper feature selection with the learner specified by the column, and then building classification models using the learner specified by the row. The first table, Table 2, includes the results averaged across all nine datasets. In the next two tables, however, these datasets are broken into three groups based either on their difficulty of learning (Table 3) or their balance level (Table 4). Which datasets are assigned to each of the three levels of these two categories is explained in Section 4, specifically in Table 1. Within each table, the best value in a given row (e.g., choice of classification learner and (for Tables 3 and 4) group of datasets) is printed in **bold**, while the worst value is printed in *italics*. This way, the reader can easily identify the best choice of internal (wrapper) learner,

Classification Learner	Wrapper Learner		
	5-NN	LR	NB
5-NN	0.78887	*0.78702*	**0.79445**
LR	*0.77689*	0.80298	**0.80562**
NB	*0.72315*	0.78442	**0.80261**

Table 2: Results Across All Datasets

Dataset Difficulty	Classification Learner	Wrapper Learner		
		5-NN	LR	NB
Easy	5-NN	**0.93952**	0.92993	*0.92943*
	LR	*0.92949*	0.94275	**0.94369**
	NB	*0.92569*	0.94271	**0.94641**
Moderate	5-NN	0.79467	*0.78433*	**0.80851**
	LR	*0.77821*	**0.82730**	0.81259
	NB	*0.66420*	0.75102	**0.80576**
Hard	5-NN	*0.63242*	**0.64680**	0.64540
	LR	*0.62297*	0.63888	**0.66056**
	NB	*0.57956*	**0.65952**	0.65566

Table 3: Results Broken Down by Dataset Difficulty

given the choice of external (classification) learner.

In Table 2, we see the results averaged across all nine datasets. The most striking observation here is that NB is the best learner for selecting the features, regardless of which learner will be used for building the final classification model: that is, no matter the choice of external learner, the best internal learner is NB. In addition, 5-NN is particularly bad at selecting the internal features: it is the worst learner for this in all cases where it isn't also used as the external learner. From these results, we can see that contrary to popular opinion, the best models are not always built by selecting a single learner and using it both inside and outside the wrapper: once the choice of external wrapper has been made, the best learner to use inside the wrapper is always NB. However, it is possible that these results will vary based on the properties of the datasets being used. For this reason, we consider the results when breaking the datasets down into categories based on two parameters: dataset difficulty and dataset balance.

Table 3 contains the results broken down by dataset difficulty (that is, difficulty of learning). We see that although these are similar to the results across all datasets, there are some notable differences. In particular, there is no difficulty level where NB is the best choice of internal learner across all choices of external learner. Instead, for the Easy and Moderate datasets, NB is the best internal learner for only two of the three choices of external learner: the 5-NN learner is the best wrapper learner when 5-NN is used as the classification learner for the Easy datasets, and the LR learner is the best wrapper learner when LR is used as the classification learner for the Moderate datasets. Thus, for those two groups of datasets, two of the three learners work best when the internal and external learners are matched. The Hard datasets show the opposite: the optimal value is never found when the internal and external learners match. Instead, LR is the best internal learner as long as it is *not* the external learner,

Dataset Balance	Classification Learner	Wrapper Learner		
		5-NN	LR	NB
Balanced	5-NN	**0.79513**	0.78986	*0.76887*
	LR	0.78918	**0.81020**	*0.78516*
	NB	*0.69218*	0.75044	**0.77762**
Slightly Imbalanced	5-NN	*0.78699*	0.81194	**0.81786**
	LR	*0.78326*	0.82448	**0.83570**
	NB	*0.74936*	**0.82686**	0.81589
Imbalanced	5-NN	0.78449	*0.75926*	**0.79662**
	LR	*0.75823*	0.77426	**0.79598**
	NB	*0.72792*	0.77596	**0.81430**

Table 4: Results Broken Down by Dataset Balance

but when it is the external learner, NB is best as the internal learner. From this, we can say that whether "matching the learner inside and outside the wrapper" is an optimal strategy may depend on the difficulty of the dataset being studied.

In addition, looking at the learners which gave the worst results inside the wrapper helps explain some of the effects we saw across all datasets. In particular, for the Hard datasets, 5-NN is always the worst learner to use inside the wrapper (regardless of the learner used for final classification), but for the Easy and Moderate datasets, it is only worst when the LR and NB learners are used for external classification. As noted previously, for the Easy datasets it is actually best when 5-NN is also used for external classification (and in this case, NB is the worst choice of internal learner). On the Moderate datasets, however, 5-NN is neither the best nor worst internal learner when 5-NN is the external learner; instead, these positions are held by NB and LR, respectively. Thus, we see that the identity of the worst learner can also vary depending on the inherent dataset difficulty, and that using 5-NN as the external learner can lead to particularly unstable results (with a different "worst internal learner" for all three levels of dataset difficulty).

The results broken down by level of dataset balance are presented in Table 4. Here, we see perhaps the strongest indicator that dataset properties (in this case, balance) can affect whether matching the learner inside and outside the wrapper will help optimize performance. When considering the most imbalanced datasets, the best internal learner is always NB, as we saw across all datasets. However, when we consider only the Balanced datasets, we find that matching the internal and external learners is always the optimal choice. That is, when 5-NN is the external learner, 5-NN is the best internal learner; when LR is the external learner, LR is the best internal learner; and when NB is the external learner, NB is the best internal learner. Conversely, when we consider the Slightly Imbalanced datasets, matching the external and internal learners is *never* the optimal strategy: instead, NB is the best internal learner when 5-NN or LR are the external learner, but LR is the best internal learner when NB is the external learner. These differences suggest that the best choice of internal learner is highly affected by dataset balance level, or at least by the specifics of the datasets being studied.

Although the optimal choice of internal learner shows a

great deal of variation across different balance levels, the worst choice is relatively constant: 5-NN is the worst internal learner when LR or NB is the external learner across all three balance levels (except for the Balanced data, where NB is worse than 5-NN as the internal learner when LR is the external learner). With 5-NN as the external learner, however, the worst choice of internal learner depends on the balance level: NB when considering Balanced datasets, 5-NN when considering Slightly Imbalanced datasets, and LR when considering Imbalanced datasets. Overall, these results are similar to those found when considering the three levels of dataset difficulty: 5-NN is consistently a bad choice of internal learner whenever it is not also being used as the external learner, but when 5-NN *is* the external learner, the worst choice will depend heavily on the properties of the dataset.

6 Conclusion

Due to the problem of high dimensionality, dimensionality reduction techniques such as feature selection are an important aspect of studying bioinformatics datasets. While many forms of feature selection exist, there has been insufficient research on using wrapper-based feature selection in the domain of bioinformatics, and thus we chose to study this using three learners (5-NN, LR, and NB) and nine bioinformatics datasets. Although most earlier studies of wrapper-based feature selection have matched the learner inside and outside the wrapper, we sought to understand whether this is always an optimal strategy, or whether other configurations of internal and external learners would yield superior classification performance for certain scenarios. Thus, our nine datasets include a wide range of both difficulty of learning and of balance level, and we considered the performance of the nine models (three choices of wrapper learner and three choices of classification learner) across different levels of these two parameters, as well as across all datasets together.

Our results show that over all datasets, NB is the best choice of wrapper learner regardless of which learner is used for the final classification model. However, when looking more closely at each group of datasets, we find that this result only holds when considering the most imbalanced datasets. With the most difficult datasets, LR is the best internal learner except when LR itself is also the external learner, and for the other two difficulty levels, NB is the best internal learner for only two of the three choices of external learner — the third choice (5-NN or LR, depending on the difficulty level) works best when matched with itself. As for the different levels of balance, we find that on the most balanced data, it is always best to match the same learner inside and outside the wrapper. However, on the slightly imbalanced datasets, rather than finding results halfway between the balanced and imbalanced results, we find that NB is the best wrapper learner *except* when NB is used as the classification learner.

Future work can extend these results to a wider range of learners and bioinformatics datasets, in order to validate these findings and discover how broadly these trends can be applied.

References

Chanrion, M.; Negre, V.; Fontaine, H.; Salvetat, N.; Bibeau, F.; Grogan, G. M.; Mauriac, L.; Katsaros, D.; Molina, F.; Theillet, C.; and Darbon, J.-M. 2008. A gene expression signature that can predict the recurrence of tamoxifen-treated primary breast cancer. *Clinical Cancer Research* 14(6):1744–1752.

Dittman, D. J.; Khoshgoftaar, T. M.; Wald, R.; and Van Hulse, J. 2010. Comparative analysis of DNA microarray data through the use of feature selection techniques. In *Ninth IEEE International Conference on Machine Learning and Applications*, 147–152.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: An update. *SIGKDD Explorations Newsletter* 11(1):10–18.

Inza, I. n.; Larrañaga, P.; Blanco, R.; and Cerrolaza, A. J. 2004. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine* 31(2):91–103.

Kohavi, R., and John, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2):273–324.

Leung, Y., and Hung, Y. 2008. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 99(1).

Lewis, D. 1998. Naive (bayes) at forty: The independence assumption in information retrieval. In Nédellec, C., and Rouveirol, C., eds., *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*. Springer. 4–15.

Pawitan, Y.; Bjohle, J.; Amler, L.; Borg, A.-L.; Egyhazi, S.; Hall, P.; Han, X.; Holmberg, L.; Huang, F.; Klaar, S.; Liu, E.; Miller, L.; Nordgren, H.; Ploner, A.; Sandelin, K.; Shaw, P.; Smeds, J.; Skoog, L.; Wedren, S.; and Bergh, J. 2005. Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Research* 7(6):R953–R964.

Peng, H.; Long, F.; and Ding, C. 2005. 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.

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

Van Hulse, J.; Khoshgoftaar, T. M.; and Napolitano, A. 2011. A comparative evaluation of feature ranking methods for high dimensional bioinformatics data. In *2011 IEEE International Conference on Information Reuse and Integration*, 315–320.

Wang, Y.; Tetko, I. V.; Hall, M. A.; Frank, E.; Facius, A.; Mayer, K. F. X.; and Mewes, H. W. 2005. Gene selection from microarray data for cancer classification–a machine learning approach. *Computational Biology and Chemistry* 29(1):37–46.

Witten, I. H.; Frank, E.; and Hall, M. A. 2011. *Data Mining: Practical machine learning tools and techniques*. Burlington, MA: Morgan Kaufmann, 3rd edition.

Xiong, M.; Fang, X.; and Zhao, J. 2001. Biomarker identification by feature wrappers. *Genome Research* 11(11):1878–1887.

Zhu, Z.; Ong, Y.-S.; and Dash, M. 2007. Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37(1):70–76.