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

Classification Performance of Rank Aggregation Techniques for Ensemble Gene Selection

David J. Dittman, Taghi M. Khoshgoftaar, Randall Wald, and **Amri Napolitano**
Florida Atlantic University
ddittman@fau.edu, khoshgof@fau.edu, rdwald@gmail.com, amrifau@gmail.com

Abstract

A very promising tool for data mining and bioinformatics is ensemble gene (feature) selection. Ensemble feature selection is the process of performing multiple runs of feature selection and then aggregating the results into a final ranked list. However, a central question of ensemble feature selection is how to aggregate the individual results into a single ranked feature list. There are a number of techniques available, ranging from simple to complex; the question is which one to choose. This paper is a comprehensive study on the use of nine different rank aggregation techniques for building classification models to use gene microarray data for distinguishing between cancerous and non-cancerous cells (or between patients who did or did not respond well to cancer treatment). The techniques are tested using an ensemble with twenty-five feature selection techniques and fifty iterations along with eleven bioinformatics datasets and five learners. Our results show that Lowest Rank is the worst performing aggregation technique by a clear margin. The other techniques perform similarly well and a simple technique (e.g., Mean aggregation) is preferable due to computation time and the limited possible benefit of a more complex technique. To our knowledge there has never been a study this intensive on the classification abilities of rank aggregation techniques in the field of bioinformatics.

Introduction

Dimensionality reduction techniques have become commonplace in bioinformatics research. These techniques allow researchers to focus on the specific data that is needed for their work. However, there are a number of techniques to choose from, each with their own abilities and biases. One of the possible solutions to this decision is ensemble gene selection.

Ensemble gene or feature selection is the process of performing multiple runs of feature selection (choosing an optimum subset of features and performing any subsequent analysis on only those features) and then aggregating those results in to a single feature subset. There are a number of benefits of ensemble feature selection including: more stable feature lists, comparable or superior classification results compared to individual techniques, and a reduction of bias in the decision process.

However, one important aspect of ensemble feature selection is how to aggregate the results of the individual runs of feature selection. There are a number of techniques to choose from, ranging from simple to complex. The question is which of these techniques is the most appropriate for optimum results.

This paper is a thorough study of nine rank aggregation techniques and their classification performance. To compare these, we use an ensemble of twenty-five feature selection techniques on fifty iterations (each technique is used twice) with eleven bioinformatics datasets (specifically, gene microarray datasets used for distinguishing between either cancerous and non-cancerous cells or response and non-response to cancer treatment). Additionally, we also use five learners for building classification models. Our results show that in general, there is little difference between the techniques, with the exception of Lowest Rank which was clearly the worst. This allows us to state that the effect of the choice of rank aggregation technique is minimal and that a more simple and efficient technique (e.g., Mean aggregation) is preferable to a more complex and computationally expensive technique for the potential small benefit.

This paper is organized as follows. Related Works contains some background information regarding our topic. Rank Aggregation Techniques contains information on the nine rank aggregation techniques used in this paper. Methodology contains the process of our experiments. The Results section describes what we observed during our experiments. Lastly, the Conclusion section presents our findings and future work.

Related Works

The use of ensembles has been most frequently applied to the creation of learners for building inductive models. It has been shown that these ensemble learners are competitive with other learners and in some cases are superior. This has been found to be true within the domain of bioinformatics (Yang et al. 2010). Recently, there have been studies applying the ensemble concept to the process of feature selection (Abeel et al. 2010). The benefits of using ensemble feature selection include more stable feature lists and comparable or superior classification results when compared to

individual (e.g., non-ensemble) feature selection (Awada et al. 2012).

However, with ensemble feature selection comes a decision of how to aggregate the results. A number of different rank aggregation techniques have been proposed in the literature: some are simple (Mean, Median, Highest Rank, Lowest Rank), and some are less so. Recent work in the area of rank aggregation techniques has centered around developing unique and innovative approaches. These new techniques can focus on different aspects of the ranking process, including comparing results to randomly generated results (Kolde et al. 2012), giving more weight to top ranking features (Haury, Gestraud, and Vert 2011), or combining two known techniques to enhance each other (Aslam and Montague 2001). While there has been work focusing on comparing a large number of rank aggregation techniques (Wald et al. 2012), the focus of that work was on the similarity of the selected feature subsets, not the classification results. Additionally, previous research has shown that the choice of aggregation technique can affect classification results (Wald, Khoshgoftaar, and Dittman 2012).

Rank Aggregation Techniques

Many techniques are used throughout the literature to combine multiple ranked lists into one final product; these are referred to as rank aggregation techniques. In this paper, we study nine such techniques: Mean, Median, Highest Rank, Lowest Rank, Stability Selection, Exponential Weighting, Enhanced Borda, Round Robin, and Robust Rank Aggregation. All of these techniques assume that the ranked lists being combined assign a value to each feature, from 1 to N (for N features), where the best feature is assigned number 1, the second-best feature is 2, and so on until the worst feature is assigned N. Unless otherwise noted, the ranked lists produced by the ensemble techniques will also assign values to each feature such that lower values are better.

While some rank aggregation techniques employ complex algorithms to assign values to the features based on their position in the various lists being combined, some are relatively straightforward. Mean aggregation simply finds the mean value of the feature's rank across all the lists and uses this as that feature's value. Similarly, Median aggregation finds the median rank value across all the lists being combined, using the mean of the middle two values if there are an even number of lists. Highest Rank and Lowest Rank use related strategies: either the highest (best, smallest) or lowest (worst, largest) rank value across all the lists is assigned as the value for the feature in question. In all cases, once each feature has been given a single value based on the mean, median, highest, or lowest value, all features are ranked based on these new values. Note that for all four of these it is possible for two features to end up tied, even if this was not the case in any of the lists being combined; this tie is resolved randomly if necessary.

Stability Selection is based on a very simple principle: a feature is good if it appears towards the top of many of the ranked lists being aggregated. To this end, a threshold is chosen. This is often the threshold which will be used for selecting the features for downstream classification, but may be any appropriate value. For each list where the feature in question meets or exceeds that threshold, that feature is given a single point. For those lists where it fails to meet the threshold, it is given zero points. This calculation is performed for all of the ranked lists, and each feature is given all the points it deserves. Finally, the features are ranked from most to least points (Haury, Gestraud, and Vert 2011).

While Stability Selection performs its task of discovering those features which are most often towards the top of the list, it fails to account for how close to the top they are, and penalizes features which are just slightly under the threshold. A refinement upon this procedure is Exponential Weighting, which assigns points based on $e^{-r/s}$, where r is the feature's rank and s is the threshold being used. This satisfies the same goal as Stability Selection, while allowing for additional weight to be allocated as appropriate. As with Stability Selection, those feature which collect the most value are placed at the front of the final list (Haury, Gestraud, and Vert 2011).

It has been shown that using the Borda count (Aslam and Montague 2001) as the basis of feature ranking is mathematically equivalent to simple Mean Aggregation. However, a variation known as Enhanced Borda expands upon the original method by multiplying each feature's Borda score ($\sum_L N - r_l$) by its Stability Selection score (number of times the feature meets or exceeds a specified threshold). This ensures that features which are frequently above the threshold get extra weight, while features also get more weight based on how high up they are on the list. Again, as with Borda (and Stability Selection), higher values are better in the results.

The Round Robin rank aggregation technique combines the lists in a fairly simple, although random, fashion: the lists themselves are randomly assigned an order, and then the first feature from the first list is chosen as the first feature for the final list, then the first feature from the second list, then the third list, and so on until the first feature from each list has been selected in order. After this, the second feature from each list (again examining the lists in the same order as before) are chosen. This proceeds until no features have not been found at least once. If while proceeding in this fashion a feature is found which is already on the final list, it is disregarded (and left in its existing, higher position) (Neumayer, Mayer, and Nørvåg 2011).

When combining ranked lists of features, one important question is how well each of the ranked lists performs compared to a randomly-sorted list. This is not an easy task, because it is not known in advance what the correct feature order is. However, if one assumes that most of the ranked lists are useful, and only a few are similar to the null (randomly-sorted) list, some progress can be made. The starting point of the Robust Rank Aggregation algorithm is examining how high a given feature scored on the various ranked lists. These values are collected into a so-called *rank order*, ordered from best to worst. If a feature is particularly useful, the predominance of these values should be towards the smaller (better) end, while only those ranked lists which are similar to the null list will give values that are randomly distributed along the range. Given each of these val-

ues, the algorithm finds the probability that if all the ranked lists were random, the values would be smaller (better) than is actually seen in the real ranking. It is expected that this probability will be small for good features, and so smaller values of this metric are better (Kolde et al. 2012).

Methodology

Datasets

Table 1 contains the list of datasets used in our experiment along with their characteristics. The datasets are all DNA microarray datasets acquired from a number of different real world bioinformatics, genetics, and medical projects. As some of the techniques require that there be only two classes, we can only use datasets with two classes (in particular, either cancerous/noncancerous or, in the case of the mulligan-r-pd and mulligan-r-nr datasets, relapse/no relapse following cancer treatment). The datasets in Table 1 show a large variety of different characteristics such as number of total instances (samples or patients) and number of features. The fourth through tenth columns show the classification performance on these datasets when building models without feature selection. These are used to show that in addition to having many thousands of features, these datasets are notable for being difficult to model (such that models without feature selection do not perform well), which suggests that they may also be difficult to select features from. Six learners are used to generate these no-feature-selection models, four of which (5-NN, MLP, Naïve Bayes, and SVM) are discussed in more detail under the Classification section. C4.5D and C4.5N are both variants of the C4.5 decision tree algorithm, which builds a tree by deciding at each node which feature best divides the instances into sub-trees, and iterating on these sub-trees until a stopping criterion is met. The "D" and "N" versions differ in terms of their parameter settings. The "Average" column at the end contains the average performance across all six learners. For all models, performance is measured using AUC, the Area under the ROC Curve, where the ROC Curve itself is a plot of True Positive Rate versus False Positive Rate. All models were built using one run of five-fold cross-validation (see Cross Validation under Methodology). The idea behind using these difficult datasets is that they are more dependent on feature selection to improve their classification results.

Ensemble Technique

In order to throughly test the abilities of the rank aggregation technique we must use an ensemble approach. In this work we have chosen to use a hybrid approach which combines two different approaches to ensemble feature selection into a single technique. The hybrid approach (Figure 1) begins with the creation of (in this work) 50 different datasets for use in the technique. We chose 50 iterations as it is the smallest number of iterations which can accomodate the use of 25 feature selection techniques used multiple times. The datasets are created through the use of bootstrapping with replacement. This step creates diversity within the data being used for feature selection. The next step applies a different feature selection method to each of the different datasets. This step was included to ensure that there is diversity with the feature selection techniques and to eliminate any possible bias toward any of the feature selection techniques. In this work we use fifty iterations of feature selection and use an ensemble of 25 rankers (each ranker is used twice).

Feature Ranking Techniques

In this paper we use an ensemble of twenty-five filter-based feature ranking techniques. The twenty-five feature ranking techniques can be split into three categories: Threshold-based Feature Selection (TBFS) Techniques, First Order Statistics based feature selection, and commonly-used techniques. Eleven of the techniques (Area under ROC curve, Deviance, F–Measure, Geometric Mean, Gini Index, Kolmogorov-Smirnov statistic, Mutual Information, Odds Ratio, Power, Probability Ratio, and Area Under the Precision Recall Curve) fall under the category of TBFS techniques. TBFS uses the normalized feature values as ersatz posterior probabilities and then applies various classifier performance metrics to determine the quality of the feature being examined (Dittman et al. 2011).

Seven of the techniques (Fisher Score, Fold Change Ratio, Fold Change Difference, Wilcoxon Rank Sum, Significance Analysis of Microarrays, Welch T Statistic, and Signal to Noise) belong to a family of techniques denoted as First Order Statistic based feature selection techniques (Khoshgoftaar et al. 2012). These seven techniques were combined into a single family because all seven exhibit the use of first order statistical measurements such as mean and standard deviation. Our research shows that in general this family of feature selection techniques are powerful and create very diverse feature subsets. This combination of diversity and power make them ideal to be included in ensemble feature selection.

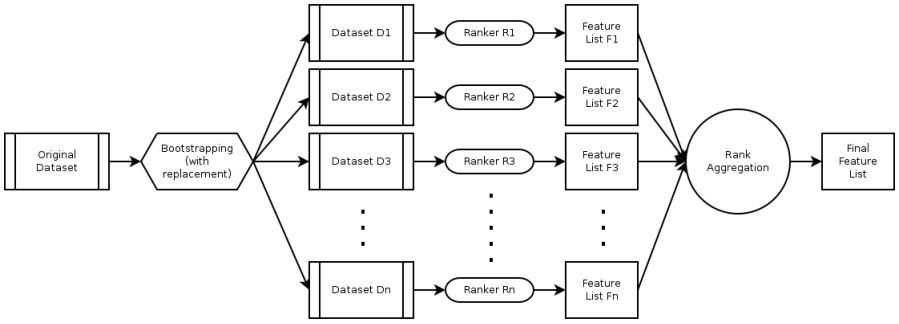
Fisher Score (Gu, Li, and Han 2011) calculates the weighted squared difference between the mean values for each class (positive and negative) compared with the overall mean value, divided by the variance for all instances. Fold Change Ratio and Fold Change Difference (Jeffery, Higgins, and Culhane 2006) find either the ratio of or the difference between the mean value of the positive and negative classes. Signal-to-noise (Dittman et al. 2011) is the ratio of the difference in mean values across the two classes divided by the sum of the standard deviations from both classes. The Welch T Statistic (Tusher, Tibshirani, and Chu 2001) is a modified version of the t-statistic which does not assume equal variance with each of the classes, while the Wilcoxon Rank Sum (Breitling and Herzyk 2005) is different from the standard t-statistic in that it makes no assumptions on whether or not the distribution is normal. Finally, the Significance Analysis of Microarrays (Tusher, Tibshirani, and Chu 2001) is the ratio of the difference of mean values from the two classes over the sum of the overall standard deviation and an exchangeability constant.

The remaining seven techniques (Chi-Squared, Information Gain, Gain Ratio, ReliefF, ReliefF-W, Symmetric Uncertainty , and SVM–RFE) are of the non-TBFS category, and are implemented in the open-source Weka machine learning toolkit (Witten and Frank 2011). Due to space lim-

Table 1: Details of the Datasets

Name	Total # of Instances	# of Attributes	AUC value						
			5-NN	C4.5D	C4.5N	MLP	Naïve Bayes	SVM	Average
colon	62	2001	0.8568	0.8420	0.8585	0.7284	0.6392	0.8398	0.79413
ovarian_mat	66	6001	0.8775	0.6163	0.6413	0.8763	0.7650	0.9613	0.78958
prostate	136	12601	0.8895	0.7651	0.8165	0.6850	0.5861	0.9512	0.78225
Brain_Tumor	90	27680	0.8053	0.7378	0.7287	0.5094	0.6372	0.9072	0.72096
lungcancer-ontario	39	2881	0.6278	0.7361	0.7597	0.7167	0.7292	0.7486	0.71968
ECML_Pancreas	90	27680	0.7973	0.6814	0.6799	0.3994	0.5000	0.9756	0.67226
mulligan-r-pd	126	22284	0.6628	0.6402	0.6471	0.5509	0.6996	0.7154	0.65265
breast-cancer	97	24482	0.7097	0.5882	0.5791	0.4770	0.5217	0.7293	0.60085
mulligan-r-nr	169	22284	0.7063	0.5146	0.5335	0.5097	0.6160	0.6783	0.59308
DLBCL-NIH	240	7400	0.5163	0.5220	0.5174	0.6727	0.6168	0.6664	0.58527
CNS	60	7130	0.5653	0.4139	0.4505	0.4646	0.5842	0.6349	0.51813

Figure 1: Diagram of Hybrid Diversity Ensemble Approach



itations we cannot elaborate on the specifics of each of the techniques; please refer to (Dittman et al. 2011) for more information.

As the goal of feature selection is to choose an optimum subset of features to perform classification, we must decide on how many of the features to use to build the classification models. Our group decided on four feature subset sizes for this experiment: 10, 25, 50, and 100. These four values span a wide range of subset sizes.

Classification

We used five different classifiers or learners to create inductive models from the features or genes chosen by the ensemble feature selection technique. These models are used to evaluate the predictive power of the genes chosen by applying them to a set of learners with varied properties. The five learners (discussed further in (Witten and Frank 2011)) work as follows: 5 Nearest Neighbor (5-NN) classifies instances by finding the five closest instances to the test instance and comparing the total weight of the instances from each class (using 1/Distance as the weighting factor). Multilayer Perceptron (MLP) builds an artificial neural network with three nodes in its single hidden layer, with 10% of the data being held aside for validating when to stop the backpropagation procedure. Naive Bayes uses Bayes' Theorem to determine the posterior probability of membership in a given class based on the values of the various features, assuming that all of the features are independent of one another. Support Vector Machines (SVM) find a maximal-margin hyperplane which cuts through the space of instances (such that instances on one side are in one class and the other side are in the other class), choosing the plane which preserves the greatest distance between each of the classes. For this study, we set SVM's complexity parameter c to 5.0 and its $buildLogisticModels$ parameter to "true" to provide proper probability estimates. Logistic Regression is a statistical technique that builds a logistic regression model to decide the class membership of future instances. All five learners use the built-in implementations in the Weka machine learning toolkit (Witten and Frank 2011), using the default parameter values unless noted elsewhere in the preceding descriptions.

Cross-Validation and Experimental Procedure

Cross-validation is a technique for training and testing models on a single dataset without the risk of overfitting. First, the dataset is divided into N folds, or subsets. Models are then trained on the first $N-1$ folds and tested on the final fold. The training and testing process is repeated until each of the folds has been the test fold, and the results for all iterations (i.e. from each of the folds, when that fold was the test fold) are collected. In this paper we use five-fold cross-validation. Additionally, we perform four runs of the five-fold cross validation so as to reduce any bias due to a

Table 2: Average Classification Performance (in AUC) of the 9 Aggregation Techniques

Learner	Subset Size	Enhanced Borda	Exponential Weighting	Highest Rank	Lowest Rank	Mean	Median	Robust Rank	Round Robin	Stability Selection
5-NN	10	0.73907	**0.74439**	*0.70145*	0.70570	0.74358	0.74135	0.72014	0.71465	0.74039
	25	0.75324	0.75625	0.74384	*0.72097*	**0.76283**	0.74973	0.73133	0.74336	0.75163
	50	0.76149	0.76168	0.75345	*0.74189*	**0.76980**	0.76642	0.75899	0.75141	0.76080
	100	0.76537	0.76853	0.76508	*0.75707*	0.76446	0.76847	0.76804	**0.77371**	0.76491
LR	10	**0.72758**	0.72525	0.71930	0.71723	*0.71557*	0.72109	0.71952	0.72401	0.72341
	25	0.70646	0.71092	0.70706	*0.70236*	0.71081	0.71489	0.71003	**0.71605**	0.70481
	50	0.69938	**0.70871**	0.69220	0.70153	0.69756	0.69118	0.69924	*0.69098*	0.69267
	100	0.69279	0.69134	*0.68000*	0.68872	0.69102	**0.69330**	0.69328	0.68027	0.69268
MLP	10	**0.75647**	0.75117	0.74400	*0.73042*	0.74316	0.74636	0.74345	0.74144	0.75235
	25	0.75452	**0.76322**	0.75343	*0.73416*	0.74418	0.75776	0.75552	0.75935	0.76011
	50	0.76516	0.76832	0.76677	*0.73987*	0.74506	0.75219	0.75678	0.76117	**0.76936**
	100	0.76904	0.76616	0.77280	*0.74300*	0.75126	0.75968	0.76472	**0.77534**	0.76997
NB	10	**0.75618**	0.75219	0.73596	0.73395	0.74271	0.75183	*0.73217*	0.73304	0.75394
	25	0.75983	0.76167	0.75676	*0.73983*	0.75330	**0.76409**	0.75214	0.75157	0.76018
	50	0.74665	0.74874	**0.76670**	*0.72049*	0.74260	0.74391	0.74739	0.76591	0.74870
	100	0.74394	0.74516	0.75299	*0.72461*	0.74038	0.74503	0.73891	**0.75365**	0.74456
SVM	10	**0.76329**	0.75616	0.75059	0.74793	0.75576	0.75705	0.74793	*0.74233*	0.75776
	25	0.76905	**0.77089**	0.75691	*0.75570*	0.76421	0.76447	0.76178	0.75616	0.76616
	50	0.76528	0.76746	0.75507	*0.74975*	0.76320	0.76090	0.75544	0.75602	**0.76932**
	100	0.76131	0.76457	0.75895	*0.74090*	0.75657	0.76125	0.75956	0.76221	**0.76500**

lucky/unlucky split.

The overall experimental procedure incorporated feature selection embedded within cross-validation, to ensure that features were selected only on training data. For each of the datasets and each of the four runs, the data was divided into five folds, and one fold was held out to serve as the training step. The remaining four folds were bootstrapped with replacement 50 times, and each of the 25 feature ranking techniques was applied to two of the bootstrapped datasets. These 50 ranked feature lists were then aggregated with one of the nine feature aggregation methods, and one of the four feature subset sizes was used to select the top features. These features were used to build a model on the four training folds using one of the five learners, and this model was finally tested on the test fold which had been held aside initially. This procedure was repeated until all folds were used as the training fold once, and until all four runs of cross-validation had been performed, and the results from the training folds were collected into a single result for the given choice of feature aggregation method, feature subset size, learner, and dataset. Finally, the entire procedure was repeated for all other choice of aggregation, subset size, learner, and dataset, and the results collected for presentation. In total we built (11 datasets $\times$ 4 runs $\times$ 5-fold cross-validation $\times$ 50 iterations) = 11,000 ranked feature lists, not counting the lists created through aggregation. In terms of inductive models we built (11 datasets $\times$ 4 runs $\times$ 5-fold cross-validation $\times$ 9 rank aggregation techniques $\times$ 4 feature subset sizes $\times$ 5 learners) = 39,600 models.

Results

This section contains the results from our experiment regarding the classification performance (in terms of AUC) of nine rank aggregation techniques. The performance was tested using an ensemble of twenty-five feature rankers along with the hybrid ensemble approach with 50 iterations (each ranker is used twice), applied toward eleven bioinformatics datasets from various genetics, biological, and biomedical experiments. Table 2 contains the results of our experiment, with all five learners presented in one table. The learner and subset size are shown in columns 1 and 2, and columns 3 through 11 contain the average classification results across the eleven datasets when keeping the learner, rank aggregation technique, and the feature subset size static. The top performing value in each row is in boldface while the worst performing value in each row is in italics.

Looking at the top performers across the five learners we see that for seven of the rank aggregation techniques (Mean, Median, Highest Rank, Stability Selection, Exponential Weighting, Enhanced Borda, and Round Robin) there is at least one combination of feature subset size and learner which will have the rank aggregation technique be the top performer. Of these seven the most common top performer is a tie between Enhanced Borda, Exponential Weighting, and Round Robin with four combinations each. The least frequent top performer was Highest Rank with only a single combination as top performer.

The remaining two rank aggregation techniques (Lowest Rank and Robust Rank) did not have a single combination of learner and feature subset size in which they were the top performing rank aggregation technique. While Robust Rank was never the top performer, it was only the worst performer once. However, Lowest Rank is by far the most frequent worst performer of the rank aggregation techniques. Lowest rank was the worst performer a total of fourteen out of twenty combinations. This leads us to say that of the

nine rank aggregation techniques one should definitely avoid Lowest Rank.

Statistical analysis (not presented due to space limitations) demonstrates that with the exception of Lowest Rank, all rank aggregation techniques perform similarly. First, a three-way ANOVA was performed, with the factors being the choice of rank aggregation, learner, and subset size. All three factors were statistically significant, meaning that at least one pair of values has different means. A Tukey's Honestly Significant Difference test performed on the first factor confirms that the Lowest Rank technique differs from the rest, while all other techniques are statistically indistinguishable. We believe this due to Lowest Rank choosing the worst value for a feature; with 25 different rankers, there is a high chance of one randomly giving a bad result . Thus, as all other rankers are indistinguishable, we see that it is preferable to choose a simple and computationally inexpensive aggregation technique such as Mean over a more complex and computationally expensive aggregation technique.

Conclusion

Ensemble gene (feature) selection is a potentially robust and powerful tool for the creation of gene lists which not only performs well in terms of classification but are stable to changes in the data. However, one of the key decisions when performing ensemble feature selection is how to aggregate the multiple ranked feature lists generated during the ensemble process. In this paper we applied nine rank aggregation techniques on an ensemble of twenty-five filter-based feature ranking techniques on eleven bioinformatics datasets. The resulting aggregated gene lists were used with five learners in order to measure their performance in terms of classification results.

It was found that for seven of the techniques (Enhanced Borda, Exponential Weighting, Highest Rank, Mean, Median, Round Robin, and Stability Selection) there is at least one combination of learner and feature subset size which will have the rank aggregation technique as the top performer. Conversely, two of the techniques (Lowest Rank and Robust Rank) did not have a combination which will place the technique as the top performer. Additionally, Lowest Rank was by far the most frequent worst performer and it is recommended not to use this technique for rank aggregation.

The results between the rank aggregation techniques have very little variance. Statistical tests confirm that other than Lowest Rank (which performs worst), all rank aggregation techniques are statistically indistinguishable. This allow us to state that the effect of the choice of rank aggregation technique is minimal and it is recommended to choose a simple and efficient technique over a more complicated one.

Future work in this area will focus on narrowing the domain of the datasets to those of specific purposes (patient response prediction to a drug, tumor identification, etc.). This is to determine if the variability of the techniques change based on a more specific data source.

References

Abeel, T.; Helleputte, T.; Van de Peer, Y.; Dupont, P.; and Saeys, Y. 2010. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics* 26(3):392–398.

Aslam, J. A., and Montague, M. 2001. Models for metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, 276–284. New York, NY, USA: ACM.

Awada, W.; Khoshgoftaar, T. M.; Dittman, D.; Wald, R.; and Napolitano, A. 2012. A review of the stability of feature selection techniques for bioinformatics data. In *2012 IEEE 13th International Conference on Information Reuse and Integration (IRI)*, 356–363.

Breitling, R., and Herzyk, P. 2005. Rank-based methods as a non-parametric alternative of the t-statistic for the analysis of biological microarray data. *Journal of Bioinformatics and Computational Biology* 3(5):1171–1189.

Dittman, D. J.; Khoshgoftaar, T. M.; Wald, R.; and Hulse, J. 2011. Feature selection algorithms for mining high dimensional dna microarray data. In *Handbook of Data Intensive Computing*. Springer New York. 685–710.

Gu, Q.; Li, Z.; and Han, J. 2011. Generalized fisher score for feature selection. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 266–273. AUAI Press.

Haury, A.-C.; Gestraud, P.; and Vert, J.-P. 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE* 6(12):e28210.

Jeffery, I.; Higgins, D.; and Culhane, A. 2006. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics* 7(1):359.

Khoshgoftaar, T. M.; Dittman, D. J.; Wald, R.; and Fazelpour, A. 2012. First order statistics based feature selection: A diverse and powerful family of feature selection techniques. In *Proceedings of the 11th International Conference on Machine Learning and Applications: Health Informatics Workshop*, 151–157. ICMLA.

Kolde, R.; Laur, S.; Adler, P.; and Vilo, J. 2012. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics* 28(4):573–580.

Neumayer, R.; Mayer, R.; and Nørvåg, K. 2011. Combination of feature selection methods for text categorisation. In Clough, P.; Foley, C.; Gurrin, C.; Jones, G.; Kraaij, W.; Lee, H.; and Mudoch, V., eds., *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 763–766.

Tusher, V. G.; Tibshirani, R.; and Chu, G. 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences* 98(9):5116–5121.

Wald, R.; Khoshgoftaar, T. M.; Dittman, D. J.; Awada, W.; and Napolitano, A. 2012. An extensive comparison of feature ranking aggregation techniques in bioinformatics. In *2012 IEEE 13th International Conference on Information Reuse and Integration*, 377–384.

Wald, R.; Khoshgoftaar, T. M.; and Dittman, D. 2012. Mean aggregation versus robust rank aggregation for ensemble gene selection. In *2012 11th International Conference on Machine Learning and Applications*, volume 1, 63–69.

Witten, I. H., and Frank, E. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.

Yang, P.; Hwa Yang, Y.; B Zhou, B.; and Y Zomaya, A. 2010. A review of ensemble methods in bioinformatics. *Current Bioinformatics* 5(4):296–308.