2011 IEEE International Conference on High Performance Computing and Communications

A Dilemma in Assessing Stability of Feature Selection Algorithms

Salem Alelyani†, Zheng Zhao‡, Huan Liu†

†Department of Computer Science and Engineering, Arizona State University, USA
‡SAS Institute Inc., USA
salem.alelyani@asu.edu,alan.zhao@sas.com,huan.liu@asu.edu

***Abstract*—In realm, feature selection is an effective means for handling high-dimensional data that becomes increasingly abundant. The stability of a feature selection algorithm is becoming crucial for determining the fitness of the algorithm. Below, we review existing methods of stability assessment and analyse how they assess the stability of a feature selection algorithm. A common approach is to evaluate the similarity between the selected subsets of features produced by that algorithm over different training samples or over distributed datasets. We point out challenges facing the existing evaluation methods and suggest how to improve stability assessment of feature selection algorithms.**

Keywords: feature selection, distributed datasets, algorithm's stability, stability assessment, Jaccard Index.

I. Introduction

The rapid development of high-throughput technologies has greatly advanced researchers' capability for collecting data enabling them to solve problems which were inextricable in the past. In the process of data mining and machine learning, high dimensional data sets accumulate and challenge existing learning techniques [17]. Given a data set of very high dimensionality the performance of existing learning algorithms usually degenerates in accuracy, efficiency and model interpretability. The problem is known as the "curse of dimensionality" [1], and to address the problem, feature selection techniques have been developed to reduce dimensionality by selecting relevant original features, and eliminating the irrelevant and redundant ones [16, 9]. In the last three decades, a large number of feature selection algorithms has been developed, and feature selection techniques have been successfully applied in various domains, including bioinformatics [21], pattern recognition [14], text mining [7], and so on. The high volume of existing feature selection approaches necessitates effective evaluation techniques to compare algorithms, so that proper ones can be chosen to serve the users' requirements. In most of the work, two major criteria are considered: accuracy and efficiency [24, 9, 18, 15, 21, 10]. In recent years, researchers gradually realized that stability is also crucial factor for evaluating feature selection algorithms [11, 12]. For instance, in genetic analysis, given data sets generated from different experiments on the same cancer cell line, researchers expect a feature selection algorithm to select similar gene sets. And any inconsistent results may cause confusion, lowering researchers' confidence on the biological relevance of selected genes [2]. Another example is in case of distributed datasets where the algorithm should produce robust results across these distributed data. Recently, stability assessment has received much attention and has become a fast-growing area in feature selection research.

Stability of a feature selection algorithm refers to the insensitivity of an algorithm to various data perturbations, which are usually caused by noise. The existence of noise is ubiquitous; therefore, a good feature selection algorithm should be robust to noise, and can return stable results, obtaining only relevant features. Given two sample sets generated by perturbing the original data, the stability of a feature selection algorithm can be measured by evaluating the similarity of the feature lists obtained by applying the algorithm on the two sample sets [11]. Fig. 1 shows a representative process for feature selection stability assessment that contains four key steps. (1) Given a data set X, one first generates l sample sets, $\mathcal{X} = \{X_1, ..., X_l\}$, either by random sampling or l-fold cross-validation. (2) A feature selection algorithm is applied to each sample set and selects features that result in l feature list, $\mathcal{R} = \{R_1, \ldots, R_l\}$. (3) Various similarity measures are applied to evaluate the pairwised similarity between the obtained features lists, which results in a similarity matrix $\mathbf{S}$. (4) The final stability estimation is computed by averaging overall obtained pairwised similarity. Among the four steps, step (3) is the pivot component of the process. And currently, most existing work on feature selection stability assessment is devoted to designing effective measurements to evaluate the similarity of two given feature lists [13, 2, 20, 22].

Unlike most existing work, in this paper, we focus on step (2) of the process, and study how to generate sample sets in a sensible way, so that the thereafter steps of the process can produce meaningful results. The motivation of this work can be shown by the following example. Assume the original data is X. And based on X, we generate two data sets X_1 and X_2. Among the two newly generated data sets, X_1 is very similar to X, while X_2 is very different to X. Given a feature selection algorithm $f(\cdot)$, whose stability is unknown, we can apply the algorithm on X, X_1 and X_2, generating three feature lists R, R_1 and R_2. Let $< \cdot >$ be

978-0-7695-4538-7/11 $26.00 © 2011 IEEE
DOI 10.1109/HPCC.2011.99

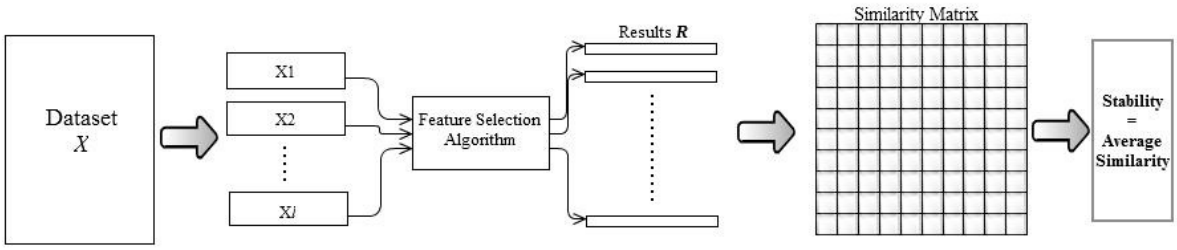

Figure 1. The process for assessing the stability of a feature selection algorithm.

a measurement assessing the similarity of two feature lists. Using $< \cdot >$, we can generate two results: $< R, R_1 >$, and $< R, R_2 >$. Let us further assume that $< \cdot >$ returns a small value in both cases, which means, neither R_1, nor R_2 is similar to R. Our question is that, should we draw the same conclusion no matter if we are given $< R, R_1 >$ or $< R, R_2 >$? The answer is obviously "no".

When $< R, R_1 >$ is small, we know that $f(\cdot)$ is unstable. Since the difference between X and X_1 is small, in this case it is reasonable for us to require a stable feature selection algorithm to generate similar results. However, if we are only given $< R, R_2 >$, it is hard for us to draw any conclusion. Since X_2 is very different to X, in this case, even a stale feature selection algorithm may generate very different feature list, due to the fact that the target concept contained by the two data sets may be completely different.

Most existing work implicitly assumes the sample sets generated in step (2) are of little difference. However, this assumption may not hold in many real-world applications. As mentioned above, given a data X, the l-folds, $X_1, ..., X_l$, are usually generated via either random sampling or l-folds cross-validation. However, we noticed that neither of the two methods can guarantee the l sample sets are of small difference. And when the differences among $X_1, ..., X_l$ are actually big, any conclusion drawn based on the output of the process loses its foundation and may no longer correct.

In this paper we urge the importance of considering data variance in the process of stability assessment for feature selection algorithms and study how to effectively measure and control the variance of the generated sample sets. To the best of our knowledge, this paper forms the first attempt that jointly considers both sample sets' similarity and feature list similarity in stability assessment for feature selection algorithms. The remaining content of the paper is organized as follows. In Section 2, we review related work. In Section 3, we develop effective methods to measure and control the variance of the samples sets generated for stability assessment. We discuss our extensive experiments in Section 4. Finally, draw conclusion in Section 5.

II. Related Work

Most existing work for feature selection stability assessment focuses on designing effective measurements to evaluate the similarity of two given feature lists. In general, different similarity measurements fall into four categories: index based methods, weight based methods, rank based methods, and feature similarity based methods [12, 25]. In [12], three measurements are used to measure the similarity between two feature lists, which include Jaccard indx, Pearson's correlation, and Spearman's correlation. The formulations for the three measurements are defined as below:

$$S_J(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|},$$

$$S_P(R_i, R_j) = \frac{\sum_k (w_{k,R_i} - \mu_{w,R_i})(w_{k,R_j} - \mu_{w,R_j})}{\sqrt{\sum_k (w_{k,R_i} - \mu_{w,R_i})^2 \sum_k (w_{k,R_j} - \mu_{w,R_j})^2}},$$

$$S_S(R_i, R_j) = 1 - 6 \sum k \frac{(r_{k,R_i} - r_{k,R_j})^2}{m(m^2 - 1)},$$

In the above equations, R_i and R_j are the two feature lists, S_J, S_P, and S_S are similarity measures derived from the Jaccard index, Pearson's correlation and Spearman correlation, respectively. w_{k,R_i} denotes the feature weight of the k-th feature in feature list R_i, μ_{w,R_i} denotes the feature weight mean of the features in R_i, r_{k,R_i} denotes the rank of the k-th feature in feature list R_i, and m is the total number of features. Among the three measurements, the Jaccard index

deals with an index of selected features, and is an index based method. The other two deal with a feature's weight and rank, and are weight-based and rank-based methods, respectively. Jaccard index, aims to evaluate the amount of overlap between two set of feature indices, while Pearson's and Spearman's correlations aim to measure the consistency of weights or ranks of the features in the two lists. When l feature lists are given, the stability of a feature selection algorithm can be inferred from all feature list similarities via the following equation:

$$S_{(J,P,S)}(\mathcal{R}) = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} S_{(J,P,S)}(R_i, R_j)$$

In [6], the authors propose average normal hamming distance (ANHD) to assess stability. Similar to Jaccard index, ANHD evaluates stability based on the indices of the selected features. In [22], the authors proposed the consistency family measurements, which aim to evaluate the stability by considering the frequency of the features in the feature lists. Other index based similarity measurements, such as Dice-Sorensen's index and Tanimoto distance metric, have also been used to evaluate the similarity of two feature lists [26, 20]. Feature similarity based methods have also been developed by researchers. In [25, 8, 19], Symmetrical Uncertainty (SU) is used to evaluate the similarity of features in the two feature lists and has been reported to be effective for evaluating feature list similarity. Current approaches do not consider data variance when assessing feature selection stability, and this may cause serious problems, when data variance is big. In the next section, we develop effective methods to measure and control the variance of the sampled data. This ensures that sensible results can be generated from existing stability measurements.

III. The Dilemma of Stability Assessment

As we described in section II, current stability measurements do not consider the influence of the variance on the results. In this paper, we first demonstrate the influence of the variation of the training datasets on the stability results by conducting an experiment using two extreme cases. The first scenario is the typical process of assessing stability that existing methods perform. We start by randomly sampling $l = 10$ different training datasets, $\mathcal{X}_1 = \{X_{11}, \ldots, X_{1l}\}$, from the original dataset X, where each subsample is 25% of the total number of samples m in X. We use 5 different datasets that vary in the number of samples m in dimensionality n; see Table I. Next, we run the algorithm $f(\cdot)$ on $\mathcal{X}_1$ which will generate l different results, R. Finally, we assess stability using Jaccard index exactly as current methodology does. The second scenario is created by generating $l = 10$ training samples $\mathcal{X}_2$. In contrast to the first scenario, $X_{21}, \ldots, X_{2l}$ are exactly the same. In other words, we randomly sample 25% of m and duplicate this subsample l times. Then, we similarly run the same algorithm $f(\cdot)$ on $\mathcal{X}_2$ as we did in the first scenario. To summarize these two scenarios, we first assumed that the datasets suffer from very huge variation between the data samples, while in the second scenario, we assumed that there is no variance between the datasets at all.

Table I
DATASETS STATISTICS

Dataset Name	Number of Samples m	Dimensionality n
CLL-SUB	111	11340
GLA-BRA	180	49151
TOX	171	5748
GLI	85	22283
PRO-CAN	171	11302

Five well-known feature selection algorithms are used in this experiment to demonstrate the consistency of the drawn conclusion. These algorithms are: ReliefF [23], ChiSquared [23], Information Gain [4], Fisher [5], and L1SVM [3]. An important question at this stage should be: which of these two scenarios' stability results should be the ground truth stability of the algorithm? In other words, there will be, for sure, two different stability results $S_{(J)}(\mathcal{R}_1)$ and $S_{(J)}(\mathcal{R}_2)$ corresponding to the first and second scenarios, respectively. Which should we consider to be the stability of the algorithm $f(\cdot)$?.

Figure 2 illustrates the stability of the above scenarios. $S_{(J)}(\mathcal{R}_1)$ is represented as $\alpha = 1$ and $S_{(J)}(\mathcal{R}_2)$ as $\alpha = 0$. As a result of the huge variance in the training samples in the first scenario, we obtained small stability in all algorithms and across different datasets with no exceptions. On the other hand, we got completely stable results in the second scenario, which means that all the generated results are always the same. Although this is an intuitive result, it provides strong evidence for the influence of the variance of the dataset on the stability results.

As a result, we can see that current measuring methods provide the assessment results that are heavily influenced by the sample variance. Now, we would like to establish the relationship between the stability measure and data variance.

A. The Impact of the Perturbation in $\mathcal{X}$ on the Stability

As we empirically prove in Section III, the results of current stability assessment methods reflect the variance of the dataset, not the exact stability of the algorithm. Here, we go further in investigating the impact of the variance by controlling the difference between the training samples $\mathcal{X} = \{X_1, \ldots, X_l\}$. We apply different amounts of perturbation $\alpha = \{10, 20, 30, 40, 90\}\,\%$ into $\mathcal{X}$. In order to do this, we randomly sample X_1 from the original dataset X. Then, we generate $X_2, \ldots, X_l$ by perturbing $\alpha\%$ of the number of samples of X_1. So, there are at least $1 - \alpha\%$ out of the total number of data samples are the same in X_1

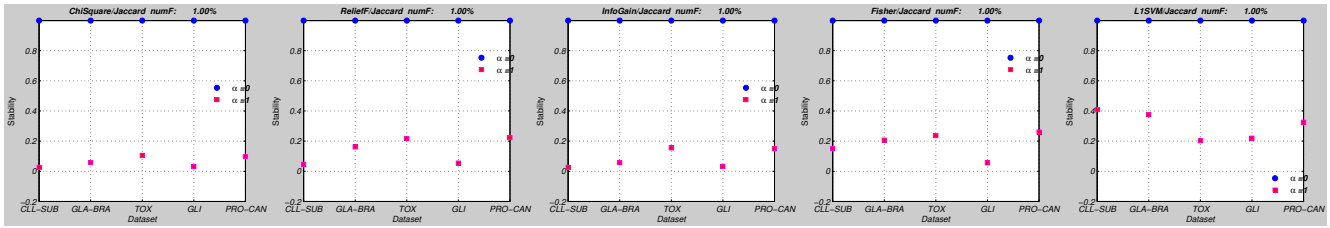

Figure 2. Stability $S_{(J)}(\mathcal{R})$ of the five methods across different datasets with two extreme cases in terms of the training samples's similarity where $\alpha = 1$ is the first scenario and $\alpha = 0$ is the second scenario.

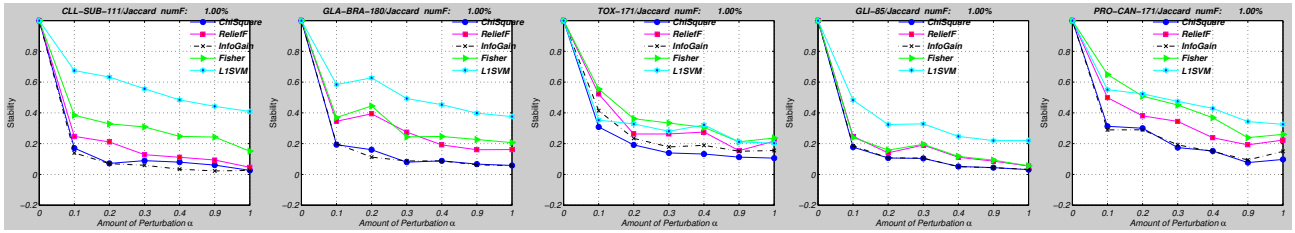

Figure 3. The stability using different amount of perturbation α. It shows the decreasing trend of the stability as α increases.

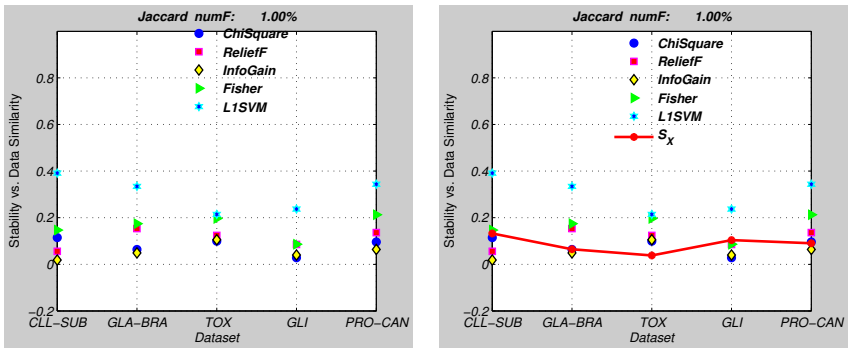

Figure 4. (a) The stability $S_{(J)}(\mathcal{R})$ of the algorithms using existing approach. (b) The pairwise similarity between training samples $S_{\mathcal{X}}$, the red line, compared with $S_{(J)}(\mathcal{R})$, the marks, where $S_{\mathcal{X}}$ is the threshold that classifies the algorithm as either stable or not .

through X_l. Figure 3 shows the stability of each algorithm across the datasets, and it clearly tends to be higher with less amount of perturbation. In this experiment, we selected 1% of the original number of features in order to get a reasonable number of relevant features.

These empirical results suggest the dependency of the stability results on the variation of the training samples. In other words, it shows the relation between the repeatability, i.e. the stability $S_{(J)}(\mathcal{R})$, of the results $\mathcal{R}$ and the similarity between the training samples $\mathbf{S}_{\mathcal{X}}$. We found that $S_{(J)}(\mathcal{R})$ is higher when α is smaller. Hence, by taking the similarity of the data samples into account, we can mitigate the effect of the data samples in the assessment of stability. This proposed methodology helps to justify and to understand the stability results.

IV. Ranking vs. Classification

After we demonstrate the influence of dataset variance on the stability results in Section III, we find that existing stability measures cannot assess the exact stability of a given algorithm without considering factors that influence the result. In fact, current methods do not assess stability, but they can only rank the algorithms according to the repeatability of the results. For illustration, Figure 4(a) shows the stability using the Jaccard index for ChiSquare, ReliefF, Information Gain, Fisher, and L1SVM. These results can only rank the algorithms according to their stability. For example, we can infer from Figure 4(a) that L1SVM is the most stable, and Fisher is the second, and so on. However, we cannot tell whether they are, in fact, stable or not since we have no clue about the variation or the similarity between training

samples. For instance, the training samples might be very similar to each other, thus, we cannot classify the algorithm as stable, owing to the fact that the cause of stability could be the the small variation in the dataset. Furthermore, the training samples might be very dissimilar. Thus, we cannot classify the algorithm to be instable as well. Similar to the first case, the instability might be caused by the huge variation in the dataset, and as a sequence, algorithms should not be expected to generate similar results. This example shows us the necessity of evaluating the similarity between training samples, in order to be able to classify the algorithms as stable or instable.

In order to infer whether a given algorithm is stable, we need to consider an important factor, which is the average pairwise similarity between the training samples. Thus, we need first to find an appropriate method to evaluate this similarity. Assume we are given $\mathcal{X}$ contains two folds X_1 and X_2, and let X_{1c_r} and X_{2c_k} to be the r^{th} and the k^{th} data point that belongs to the class c in X_1 and X_2 respectively. Then the average distance $Sim(X_{1c}, X_{2c})$ between the samples in X_1 and X_2 that belong to class c is given by:

$$Sim(X_{1c}, X_{2c}) = \frac{1}{|X_{1c}||X_{2c}|} \sum_{r=1}^{|X_{1c}|} \sum_{k=1}^{|X_{2c}|} \|X_{1c_r} - X_{2c_k}\|,$$

and the average pairwise similarity, $S_{\mathcal{X}}$, between *l-folds* in $\mathcal{X}$ is given by:

$$S_{\mathcal{X}} = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \frac{1}{|C_{i,j}|} \sum_{c \in C_{i,j}} Sim(X_{ic}, X_{jc}),$$

where $C_{i,j}$ is the intersection between the classes in X_i and X_j and $|C_{i,j}|$ is the cardinality of $C_{i,j}$. By evaluating the similarity of the training samples and the similarity of the results, we can tell whether a given algorithm is stable or not. We propose a novel approach that could simply compare the similarity of the training samples $S_{\mathcal{X}}$ against the similarity of the results $S_{(J)}(\mathcal{R})$. Then we derive the result based on this comparison. For more illustration, let X_1 and X_2 be the training samples, where $l = 2$ in this case. And let R_1 and R_2 be the generated result by the algorithm $f(\cdot)$. $f(\cdot)$ is said to be stable if $S_{(J)}(\mathcal{R}) \geq S_{\mathcal{X}}$ and unstable otherwise. As a result for this approach, the similarity values of the training samples, $S_{\mathcal{X}}$, are connected by the red line in Figure $4(b)$. With these threshold values, we can determine if an algorithm is stable or not. In other words, if $S_{(J)}(\mathcal{R})$ exceeds or equals $S_{\mathcal{X}}$, then that particular algorithm is said to be stable.

As a result, Table II illustrates which algorithm is stable or not with which dataset. According to the values ($S_{\mathcal{X}}$) in the last row in Table II, we check if $S_{(J)}(\mathcal{R}) \geq S_{\mathcal{X}}$ and determine whether an algorithm is stable or not. The stability values that are in **Bold** are the ones exceeding the threshold ($S_{\mathcal{X}}$) and thus are classified as stable. Therefore, L1SVM is always stable. Similarly, Fisher is almost always stable except for GLI. In addition, ReliefF is considered stable with GLA-BRA, TOX, and PRO-CAN. While ChiSquare is stable with TOX and PRO-CAN only. Finally, Information Gain is always unstable except with TOX dataset. In short, we empirically show that existing methods can only rank algorithms, but using the average pairwise similarity of training samples enables us to distinguish stable and instable algorithms.

V. Discussion

The assessment of the stability of the feature selection algorithms happens to be influenced by the dataset variation. The literature also suggests that there are other factors that may impact the stability such as sample size [25] and the number of selected features k [12]. These factors can be investigated independently. For example, it was shown in [12] that the stability measures can increase proportionally with the number of selected features. However, this kind of influence can be mitigated by a good estimation of k relevant features. As **(author?)** found, the increase of the stability that associated with the increase of k is mainly due to a large number of selected features that are irrelevant to the learning problem. In other words, choosing features with weight $w = 0$ to evaluate the stability is going to give higher stability since the features are added to the selected list by their sequential order. As a result of our experiment, we notice that the number of features that have $w \geq 0$ happened to be around 1% of the dimensionality. Thus, we selected 1% of features to alleviate the influence k has on the results.

Future work is to develop a feature selection method that improves selection robustness by reducing the data variance across different distributed datasets.

VI. Conclusion

In this paper, we demonstrated the dilemma of evaluating the stability of feature selection algorithms. We empirically prove that current stability assessment methodology can be heavily influenced by the variance of data samples. Therefore, the existing methods can only compare between the feature selection algorithms and rank them using stability values and cannot tell if an algorithm is stable or not, in presence of data variance. We proposed to take the training samples' similarity into account when assessing the stability. Thus, we could easily determine that the algorithm is stable or not by comparing the stability of the results with the similarity of the dataset. The stability assessment results given by our method show that some algorithms that were considered stable are actually not stable. To the best of our knowledge, this is the first work that considers the influence of the dataset variation in assessing the stability of feature selection algorithms and can provide an objective stability

	CLL-SUB	GLA-BRA	TOX	GLI	PRO-CAN
ChiSquare	0.1145	0.0638	**0.0993**	0.0284	**0.0960**
ReliefF	0.0559	**0.1535**	**0.1240**	0.0853	**0.1362**
InfoGain	0.0186	0.0491	**0.1055**	0.0401	0.0636
Fisher	**0.1468**	**0.1745**	**0.1970**	0.0861	**0.2129**
L1SVM	**0.3915**	**0.3342**	**0.2141**	**0.2373**	**0.3435**
The threshold $S_{\mathcal{X}}$	*0.1325*	*0.0648*	*0.0380*	*0.1041*	*0.0902*

Table II
THE STABILITY OF EACH ALGORITHM WITH EACH DATASET COMPARED AGAINST THE THRESHOLD WITH THE TRAINING SAMPLE SIMILARITIES $S_{\mathcal{X}}$ IN *Italic*. ALGORITHMS' STABILITY IN **BOLDFACE** ARE CONSIDERED STABLE SINCE THEY EXCEEDED THE THRESHOLD.

assessment in critical data mining and machine learning applications.

REFERENCES

[1] Richard Ernest Bellman and Rand Corporation. *Dynamic programming*. Princeton University Press, 1957.

[2] Anne-Laure Boulesteix and Martin Slawski. Stability and aggregation of ranked gene lists. *Brief Bioinform*, 10(5):556–568, 2009.

[3] P.S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Machine Learning Proceedings of the Fifteenth International Conference(ICML 98*, pages 82–90. Morgan Kaufmann, 1998.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[5] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2 edition, 2001.

[6] Kevin Dunne, Padraig Cunningham, and Francisco Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. Technical Report TCD-CD-2002-28, Department of Computer Science, Trinity College, Dublin, Ireland, 2002.

[7] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[8] Gokhan Gulgezen, Zehra Cataltepe, and Lei Yu. Stable and accurate feature selection. In *ECML/PKDD (1)*, pages 455–468, 2009.

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

[10] D. Herold, D. Lutter, R. Schachtner, A. Tome, G. Schmitz, and E. Lang. Comparison of unsupervised and supervised gene selection methods. *Conf Proc IEEE Eng Med Biol Soc*, 1:5212–5215, 2008.

[11] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms. page 8 pp., nov. 2005.

[12] Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, May 2007.

[13] Ludmila I. Kuncheva. A stability index for feature selection. In *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, pages 390–395, Anaheim, CA, USA, 2007. ACTA Press.

[14] J. Lampinen, J. Laaksonen, and E. Oja. Pattern recognition. In C.T. Leondes, editor, *Image Processing and Pattern Recognition*, volume 5 of *Neural Network Systems Techniques and Applications*, pages 1 – 59. Academic Press, 1998.

[15] Y. Y. Leung, C. Q. Chang, Y. S. Hung, and P. C W Fung. Gene selection for brain cancer classification. *Conf Proc IEEE Eng Med Biol Soc*, 1:5846–5849, 2006.

[16] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[17] Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. Feature selection: An ever evolving frontier in data mining. In *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, 2010.

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

[19] Steven Loscalzo, Lei Yu, and Chris Ding. Consensus group stable feature selection. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 567–576, New York, NY, USA, 2009. ACM.

[20] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques, 2008.

[21] Yvan Saeys, Iaki Inza, and Pedro Larraaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Oct 2007.

[22] Petr Somol and Jana Novovicov. Evaluating the stability of feature selectors that optimize feature subset cardinality, 2010.

[23] I.H. Witten and E. Frank. *Data Mining: Practical*

machine learning tools and techniques. Morgan Kaufmann Pub, 2005.

[24] Der-Shung Yang, Larry Rendell, and Gunnar Blix. A scheme for feature construction and a comparison of empirical method. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 699–704, Sydney, 1991.

[25] Lei Yu, Chris Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 803–811, New York, NY, USA, 2008. ACM.

[26] M. Zucknick, S. Richardson, and E. A. Stronach. Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods. *Stat Appl Genet Mol Biol*, 8:7, Biol 2008.