Stability of Feature Selection Algorithms

Alexandros Kalousis, Julien Prados, Melanie Hilario
University of Geneva, Computer Science Department
Rue General Dufour 24, 1211 Geneva 4, Switzerland
{kalousis, prados, hilario}@cui.unige.ch

Abstract

With the proliferation of extremely high-dimensional data, feature selection algorithms have become indispensable components of the learning process. Strangely, despite extensive work on the stability of learning algorithms, the stability of feature selection algorithms has been relatively neglected. This study is an attempt to fill that gap by quantifying the sensitivity of feature selection algorithms to variations in the training set. We assess the stability of feature selection algorithms based on the stability of the feature preferences that they express in the form of weights-scores, ranks, or a selected feature subset. We examine a number of measures to quantify the stability of feature preferences and propose an empirical way to estimate them. We perform a series of experiments with several feature selection algorithms on a set of proteomics datasets. The experiments allow us to explore the merits of each stability measure and create stability profiles of the feature selection algorithms. Finally we show how stability profiles can support the choice of a feature selection algorithm.

1 Introduction

High dimensional datasets are becoming more and more abundant in classification problems. A variety of feature selection methods have been developed to tackle the issue of high dimensionality. The major challenge in these applications is to extract a set of features, as small as possible, that accurately classifies the learning examples.

A relatively neglected issue in the work on high dimensional problems, and in general in problems requiring feature selection, is the stability of the feature selection methods used. Stability, defined as the sensitivity of a method to variations in the training set, has been extensively studied with respect to the learning algorithm itself. We propose to investigate how different subsamples of a training set affect a method's assessment of a feature's importance and consequently the final set of selected features.

The stability of classification algorithms was examined by Turney [11] who proposed a measure based on the agreement of classification models produced by an algorithm when trained on different training sets. He defined the agreement of two classification models as the probability that they will produce the same predictions over all possible instances drawn from a probability distribution $P(X)$. Note that instances are drawn from $P(X)$ and not from $P(X, C)$, the joint probability distribution of class and training instances; the underlying reason is that the agreement of two concepts—classification models—should be examined in all possible input worlds. In order to estimate stability he suggested using $m \times 2$-fold cross-validation. In each of the m repetitions of cross-validation a classification model is produced from each one of the two folds. The two models are then tested on artificial instances drawn by sampling from $P(X)$ and their agreement is computed. The final estimation of stability is the average agreement over all m runs.

Related to the notion of stability is the bias-variance decomposition of the error of classification algorithms, [4, 1]. The variance term quantifies instability of the classification algorithm in terms of classification predictions. Variance measures the percentage of times that the predictions of different classification models, learned from different training sets, for a given instance are different from the typical (average) prediction. Bias-variance decomposition is usually done via bootstrapping, where part of the data is kept as a hold-out test set and the remainder is used to create different training sets by using sampling with replacement. The final estimation of variance is also the average over the different bootstrap samples.

In both approaches described above, the predictions of the classification models are crucial in quantifying the sensitivity of classification algorithms to changes in the training sets (note that both approaches can also be used for error estimation which is then tightly coupled with the stability analysis). However when one wants to examine only feature selection algorithms without involving a classification algorithm, the above methods do not apply. Typical feature selection algorithms do not construct classification

models and thus cannot provide classification predictions. They usually output what we call a feature preference statement (for conciseness, *feature preference*); this can take the form of a subset of selected features, or alternatively of a weighting-scoring or a ranking of the features, based on which a small set of features can be selected (either by specifying a threshold or asking for a specific number of features). A classification algorithm should then be applied on the selected feature set to produce a classification model. If we used the stability estimation methods described above to the combined feature selection and classification algorithms, we would be measuring their joint sensitivity to training set variations and we would have no way to delimit the (in)stability the feature selection algorithm from that of the classification algorithm.

To address this difficulty we introduce the notion of preferential stability, i.e., the stability of the feature preferences produced by a feature selection algorithm, to quantify its sensitivity to differences in training sets drawn from the same distribution. The same approach can in fact be used to measure the preferential stability of any classification algorithm that produces models from which weightings or rankings of the features can be extracted, e.g. linear discrimination algorithms.

Stability, as introduced in [11], and the bias-variance decomposition frameworks are not able to accurately quantify preferential stability. It is possible that different training samples lead to really different feature sets which however yield the same prediction patterns. This can be especially true when the initial features have a high level of redundancy which is not handled in a principled way by the algorithms used.

The motivation for investigating the stability of feature selection algorithms came from the need to provide application domain experts with quantified evidence that the selected features are relatively robust to variations in the training data. This need is particularly crucial in proteomics applications. In mass spectrometry based diagnosis, for instance, training data (protein mass spectra) are characterized by high dimensionality and the goal is to output a small set of highly discriminatory features (protein biomarkers) on which biomedical experts will subsequently invest considerable time and research effort. Domain experts tend to have less confidence in feature sets that change radically with slight variations in the training data. Data miners have to convince them not only of the predictive potential but also of the relative stability of the proposed features or biomarkers.

The rest of the paper is organized as follows: in Section 2 we introduce measures of stability that can be applied to any feature selection algorithm that outputs a feature preference as defined above; we also show how we can empirically estimate these measures. In Section 3 we describe the experimental setup, the datasets used, and the feature selection algorithms included in the study; in Section 4 we present the results of the experiments, investigate the behavior of the different stability measures and establish the stability profiles of the chosen feature selection algorithms; in Section 5 we examine together classification performance and stability of feature preferences, and suggest how we can exploit the latter to support the choice of the appropriate feature selection algorithm; finally we conclude in Section 6.

2 Stability

The generic model of classification comprises: a generator of random vectors x, drawn according to an unknown but fixed probability distribution $P(X)$; a supervisor that assigns class labels x, to the x random vectors, according to an unknown but fixed conditional probability distribution $P(C|X)$; a learning space populated by pairs (x, c) drawn from the joint probability distribution $P(X, C) = P(C|X)P(X)$.

We define the *stability* of a feature selection algorithm as the sensitivity of the feature preferences it produces to differences in training sets drawn from the same generating distribution $P(X, C)$. Stability quantifies how different training sets affect the feature preferences.

Measuring stability requires a similarity measure for feature preferences. This obviously depends on the representation language used by a given feature selection algorithm to describe its feature preferences; different representation languages call for different similarity measures. We can distinguish three types of representation languages for feature preferences. In the first type a weight or score is assigned to each feature indicating its importance. The second type of representation is a simplification of the first where instead of weights ranks are assigned to features. The third type consists only of sets of features in which no weighting or ranking is considered. Obviously any weighting schema can be cast as a ranking schema, which in turn can be cast as a set of features by setting a threshold on the ranks or asking for a given number of features.

More formally, let training examples be described by a vector of features $f = (f_1, f_2, ..., f_m)$, then a feature selection algorithm produces either:

- a weighting-scoring: $w = (w_1, w_2, .., w_m), w \in W \subseteq R^m$,
- a ranking: $r = (r_1, r_2, .., r_m), 1 \leq r_i \leq m$,
- or a subset of features: $s = (s_1, s_2, .., s_m), s_i \in \{0, 1\}$, with 0 indicating absence of a feature and 1 presence.

In order to measure stability we need a measure of similarity for each of the above representations. To measure similarity

between two weightings w, w', produced by a given feature selection algorithm we use Pearson's correlation coefficient

$$S_W(w, w') = \frac{\sum_i (w_i - \mu_w)(w'_i - \mu_{w'})}{\sqrt{\sum_i (w_i - \mu_w)^2 \sum_i (w'_i - \mu_{w'})^2}},$$

where S_W takes values in [-1,1]; a value of 1 means that the weightings are perfectly correlated, a value of 0 that there is no correlation while a value of -1 that they are anticorrelated.

To measure similarity between two rankings r, r', we use Spearman's rank correlation coefficient

$$S_R(r, r') = 1 - 6 \sum_i \frac{(r_i - r'_i)^2}{m(m^2 - 1)},$$

where r_i and r'_i are the ranks of feature i in rankings r and r' respectively. Here too the possible range of values is in [-1,1]. A value of 1 means that the two rankings are identical, a value of 0 that there is no correlation between the two ranks, and a value of -1 that they have exactly inverse orders.

Finally we measure similarity between two subsets of features using a straightforward adaptation of the Tanimoto distance between two sets, [2]:

$$S_S(s, s') = 1 - \frac{|s| + |s'| - 2|s \cap s'|}{|s| + |s'| - |s \cap s'|}.$$

The Tanimoto distance metric measures the amount of overlap between two sets of arbitrary cardinality. S_S takes values in [0,1] with 0 meaning that there is no overlap between the two sets, and 1 that the two sets are identical.

To empirically estimate the stability of a feature selection algorithm for a given dataset, we can simulate the distribution $P(X, C)$ from which the training sets are drawn by using a resampling technique like bootstrapping or cross-validation. We opted for N-fold stratified cross-validation (N=10). In ten-fold cross-validation the overlap of training instances among the different training folds is roughly 78%. The feature selection algorithm outputs a feature preference for each of the training folds. The similarity of each pair of feature preferences, i.e. $N(N-1)/2$ pairs, is computed using the appropriate similarity measure and the final stability score is simply the average similarity over all pairs.

We want to couple stability estimates with classification error estimates in view of identifying feature selection algorithms which maximize both stability and classification performance. To this end we embed the procedure described above within an error estimation procedure, itself conducted using stratified 10-fold cross-validation. In other words, at each iteration of the cross-validated error estimation loop, there is a full internal cross-validation loop aimed at measuring the stability of feature precedences returned by the feasture selection algorithm. The outer loop provides a classification error estimate in the usual manner, while the inner loop provides an estimate of the stability of the feature selection algorithm.

3 Stability Experiments

As already mentioned, the stability of feature selection methods is of utmost importance in mass-spectra based diagnosis. Briefly, a biological sample is submitted to a mass spectrometer to produce a mass spectrum. This can be viewed as a protein profile of the sample which should be analysed to extract potential disease markers, whether individual proteins or sets of interacting proteins. To discover biomarker patterns in mass spectra, the data miner must face a number of technical challenges, foremost among which is their extremely high dimensionality. A typical mass spectrum has several thousands of features that exhibit a high degree of spatial redundancy. In order to reduce dimensionality the spectra are preprocessed, and peaks, which roughly correspond to individual proteins, are extracted. However this still leaves us with a considerable number of features. Each feature corresponds to a specific mass value, M/Z, and provides the intensity of the signal at that mass value.

We worked with three different mass spectrometry datasets: one for ovarian cancer [7], (version 8-07-02), another for prostate cancer [8] and an extended version of the early stroke diagnosis dataset used in [9]. They all involve two-class problems, diseased vs controls. Preprocessing for feature extraction consisted of baseline removal, denoising, smoothing, peak detection and peak alingment (the exact details of preprocessing are given in [6]). A short description of these datasets is given in table 1; all features correspond to intensities of M/Z values and are continuous.

For feature selection we selected Information Gain (IG), [2], ReliefF (RF), [10], and SVMRFE-[5]. Information gain is a univariate feature scoring method for nominal attributes or continuous attributes discretized using the method of [3]. ReliefF delivers a weighting of the features while taking their interactions into account; it uses all features to compute distances among training instances and the K nearest neighbors of each of the M probe instances to update feature weights. We set K to 10 and M to the size of the training set, so that all instances were used as probes. SVMRFE also takes account of feature interactions in producing a ranking, with the $P\%$ lowest ranked features being eliminated at the earliest iterations of the algorithm. In our experiments, P was set to 10% and the complexity parameter C was set to 0.5.

We also include a simple linear support vector machine to show that the same type of stability analysis can be applied to any linear classifier; here too the complexity parameter was set to 0.5. Provided that all features are normalized

dataset	# controls	#diseased	# features
ovarian	91	162	824
prostate	253	69	2200
stroke	101	107	4928

Table 1. Description of mass spectrometry datasets considered.

dataset	IG			RF		
	$\overline{S_W}$	$\overline{S_R}$	$\overline{S_S}$	$\overline{S_W}$	$\overline{S_R}$	$\overline{S_S}$
ovarian	95.53	94.67	2.93	96.97	95.37	72.95
prostate	82.47	78.19	0.91	95.72	93.99	55.29
stroke	83.87	79.39	2.68	88.06	82.30	34.10
avg	87.29	84.08	2.17	93.58	90.55	54.11
	SVMONE			SMVRFE		
	$\overline{S_W}$	$\overline{S_R}$	$\overline{S_S}$	$\overline{S_W}$	$\overline{S_R}$	$\overline{S_S}$
ovarian	93.79	84.76	45.62	NA	83.86	46.80
prostate	86.85	73.89	52.43	NA	73.23	44.84
stroke	81.74	70.33	27.21	NA	69.71	16.78
avg	87.46	76.33	41.75	NA	75.60	36.14

Table 2. Stability results for the different stability measures. $\overline{S_S}$ is computed on the feature sets of the best ten features proposed by each method.

to a common scale, the absolute values or the squares of the coefficients of the linear hyperplane can be taken to reflect the importance of the corresponding features, in effect providing a feature weighting. This is actually the assumption under which SVMRFE works; alternatively the support vector machine is equivalent to SVMRFE with a single iteration, where the ranking of the features is simply based on the absolute values or the squares of the coefficients of the support vector machine. We consider this version of support vector machines as yet another feature selection algorithm and identify it as SVMONE. The implementations of all the algorithms are those found in the WEKA machine learning environment [12].

As already mentioned the stability estimates are calculated within each training fold by a nested cross-validation loop and the final results reported are the averages, $\overline{S_W}, \overline{S_R}, \overline{S_S}$, over the ten external folds.

4 Stability Results

In table 2 we give the stability results for $\overline{S_W}$, $\overline{S_R}$, and $\overline{S_S}$, i.e., for weightings-scorings, rankings and selected feature sets, for the four different methods considered. The values of $\overline{S_S}$ depend on the imposed cardinality of the final feature set while $\overline{S_W}$ and $\overline{S_R}$ are independent of that. $\overline{S_S}$ was computed on the feature sets of the best ten features selected by each method. SVMRFE does not produce a weighting-scoring of features so the computation of $\overline{S_W}$ does not make sense in that case. For SVMONE the stability results are computed on the square values of the coefficients of the linear hyperplane found by the support vector machine.

$\overline{S_W}$ and $\overline{S_R}$ take into account the complete feature preferences produced by a method, while $\overline{S_S}$ focuses on a given number of top ranked or selected features. Thus the former two provide a global view of stability of feature preferences while the latter focuses to a more precise picture. The latter is usually of greater interest since the feature preferences are in general used to produce a restricted set of features. $\overline{S_W}$ provides a finer grain picture of stability in comparison to $\overline{S_R}$ since it is based on the actuall feature coefficients produced by a given method while the $\overline{S_R}$ uses the ranking of these coefficients. However this does not mean that the information provided by $\overline{S_W}$ is of greater value than that provided by $\overline{S_R}$, but rather the other way around. This is because again in practise we are more interested in the actual ranks of the features since based on them we will select the final set of features, differences in weights are not necessarily reflected in rank differences. A further disadvantage of $\overline{S_W}$ is that since it directly operates on the actual weights-scores produced by each method its results are not directly comparable among different methods due to possible differences in scales and intervals of the weights-scores, a problem that does not appear when we are working with the ranks. Overall the most important information is delivered by $\overline{S_S}$, when we are examining the stability of the methods for sets of selected features of given cardinality, followed by $\overline{S_R}$.

This ordering of the three measures in terms of their information content is somehow reflected on the estimated stability performances, table 2. For any method $\overline{S_W}$ gives always the highest stability estimate, followed in generally closely by $\overline{S_R}$. $\overline{S_S}$ is always considerably lower and depends on the number of features that we ask in the final feature set (remember that for the estimates of S_S in table 2 this was set to ten, later we will examine in more detail the behavior of $\overline{S_S}$ with respect to the cardinality of the final feature set). In some sense $\overline{S_W}$ and $\overline{S_R}$ provide overly optimistical estimates of feature preference stability (although in no case it can be argued that the values of the different stability measures are comparable). The reason for that can be traced on the fact that they treat all weights or ranks differences in a uniform manner. Nevertheless differences on the highest weighted or ranked features should be penalized more than differences on the lower weighted or ranked features. A fact that points to the definition and use of more

refined similarity measures that can take into account the level at which a difference appears. These similarity measures would lie conceptually between S_W, and S_R, that give equal importance to everything, and S_S, that only considers a given number of top ranked features.

We will now examine the stability performance of the four different methods considered. The clear winner is ReliefF that achieves the best performance under all stability measures for the three datasets under consideration. The performance difference is quite astonishing for $\overline{S_S}$ for which ReliefF scores 72.95%, 55.29% and 34.10% for the ovarian, prostate and stroke datasets respectively. These scores correspond to an average overlap of 8.43, 7.12 and 5.08 features out of the ten contained in the final set of selected features, among the different subsets of the training folds[1].

Information Gain appears to have a better score than SVMONE and SVMRFE for $\overline{S_W}$ and $\overline{S_R}$ (remember here that among the two it is $\overline{S_R}$ that can provide a meaningful basis for comparison of different methods). However its results are catastrophic when we consider $\overline{S_S}$, with scores of 2.93%, 0.91% and 2.68% for ovarian, prostate and stroke respectively (an average overlap of 0.56, 0.18, 0.52 features out of the ten, i.e. in average less than one common feature among the different subsets of the training folds!).

Finally the performances of SVMONE and SVMRFE are quite similar in terms of $\overline{S_R}$ a fact that can be easily explained since the ranking of features provided by SVMONE can be considered as a less refined version of the ranking provided by SVMRFE with the former being the result of a single execution of the support vector machine algorithm and the latter the result of an iterative execution where each time 10% of the lower ranked features are removed. However in terms of $\overline{S_S}$ SVMONE appears to be slightly more stable (an average overlap of 6.26, 6.87, 4.27 features out of ten for SVMONE against 6.37, 6.19, 2.87 for SVMRFE). Again the fact that SVMRFE is based on multiple iterations explains its slightly higher instability on the top ten ranked features. When the differences of the coefficients of two features are rather small and a choise is about to be made on which of the two to eliminate different training sets could result in opposite rankings for these two features thus eliminating a different feature each time.

The results of the estimation process of $\overline{S_S}$ can be very eloquently visualized providing insight not only on the stability of each individual method, but also clearly indicating which features are considered important by each method. An example of such a visualization for the prostate dataset is given in figure 1 where the cardinality of the final feature set is set to ten. In each of the graphs the x-axis corresponds

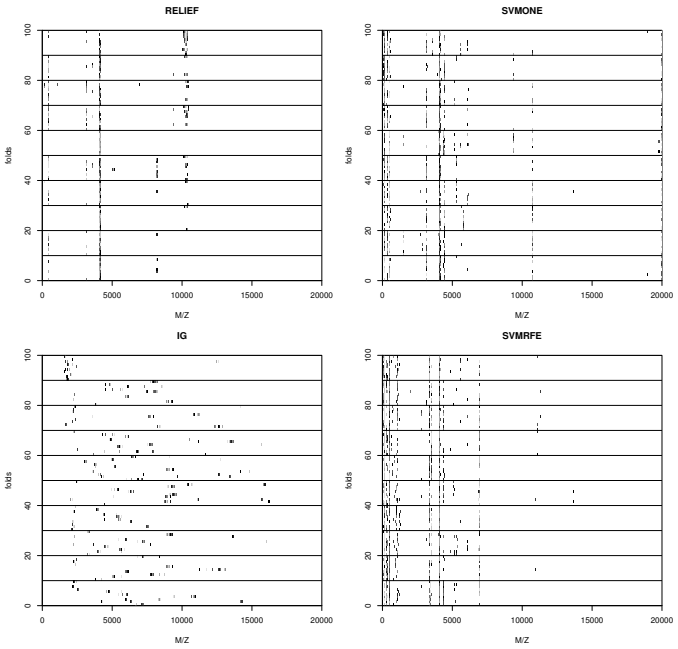

Figure 1. Stability results for the prostate dataset for selected feature sets of cardinality 10.

to the individual features. The y-axis is separated to 10 rows each one corresponding to one of the outer cross-validation folds. Within each row we find 10 rows (not visibly separated) corresponding to each of the inner cross-validation folds of the outer fold. A perfectly stable method, i.e. one that chooses always the same features, would have in its graph as many vertical lines as the cardinality of the final feature set. Each line would correspond to one selected feature. The visualization results are in perfect aggreement with the $\overline{S_S}$ estimates given in table 2. The less stable method is Information Gain with features sets selected even within the inner folds of a given outer fold being quite different (inner folds of a given outer fold share more training instances than the inner folds of two different outer folds). The other three methods are quite stable selecting very often the same features among the different inner folds.

The big differences in the stability estimates of S_R and S_S for Information Gain were puzzling. In order to see where they could be coming from we took a closer look on the weighting-scorings produced by Information Gain. It turns out that the scorings are zero, i.e. the corresponding features have a zero information gain, for a large number of features. More precisely for ovarian 35.07% of features have an information gain of zero, for prostate this goes up to 85.63%, and for stroke to 91.25%[2]. On the other hand

[1] It is easy to compute the actual number of common features when we know the S_S score and the cardinality of the final feature set simply by the definition of S_S.

[2] The presence of so many zero information gain features was the result

for ReliefF the corresponding percentages are practically zero, and for SVMONE always less than 3%. When these weightings-scorings are turned into a ranking in order to compute S_R there is a very big number of ties in the ranking of different features (in the case of information gain). The crusial element is how ties are dealt with. Originally we were assigning to all tied features their average rank. This meant that in the case of Information Gain 35.07%, 85.63% and 91.25% of the features, for ovarian, prostate and stroke respectively, had exactly the same rank; moreover these features were concentrated on the low end of the ranking. Due to the presence of a large number of features with equal ranks the final value of the $\overline{S_R}$ estimate was optimistically affected for Information Gain, moreover since this was happening on the low rank levels it was completely masking any information about the stability of the rank on the top positions.

To correct for this optimism we have chosen to break ties by assigning randomly the ranks among the tied features. For example, if below the tenth ranked feature there was a group of 20 features with exactly the same weighting-scoring then each one of them would be assigned a different rank randomly from 11 to 30. This left unaffected the $\overline{S_R}$ estimates produced for ReliefF, SVMONE, and SVMRFE, since the first two had a very low number of ties, and the latter was naturally producing a rank, but considerably lowered the stability estimates for Information Gain, with the new estimates being 91.09%, 40.74% and 20.44% for ovarian, prostate and stroke respectively, being thus more consistent with the picture that $\overline{S_S}$ is providing. However these observations still call for a more refined version of S_R that would reward similarities and penalize differences more at the top level ranks.

4.1 Stability profiles with $\overline{S_S}$

It is clear that the more interesting stability estimation is provided by $\overline{S_S}$ since it focuses only on a small subset of features, the ones selected by each method, which is actually what interest us when we are performing feature selection. To get a more precise picture of the stability performance of the different methods with respect to $\overline{S_S}$ we computed its values for different values of selected features ranging from 10 up to the cardinality of the full feature set with a step a five, figure 2. Moreover we included as a stability baseline a random feature selection that simply outputs random sets of features of given cardinality.

First remark is that ReliefF clearly dominates all other algorithms for all interesting cardinalities of the final feature sets. Information Gain has a quite bad performance for prostate and stroke, explained by the great number of zero information gain features; actually it has almost the same stability behavior as the random feature selection. In the ovarian dataset it exhibits a sharp increase of stability up to feature sets with around 300 features and then very slowly increases towards one when all features have been included. The "knot" in this curve actually corresponds to the inclusion of all features, with an information gain different than zero, to the final set of selected features. After this point features are actually added randomly. So in some sense it detects the cardinality of the most stable set of features. The same knot is also observed in the case of RelieF, quite strongly for the stroke and prostate and less for ovarian, and for SVMONE and SVMRFE in stroke. We believe that the presence of knots like these mark the inclusion of the most robust-stable features; features included later are added more or less randomly. The knots could be possibly used to determine the optimal cardinality of the most stable feature set, but this is something that needs further investigation.

SVMONE has a small advantage over SVMRFE on selected feature sets of low cardinality but their performance is indistinguishable for high cardinalities. As we move to higher cardinalities both methods add low ranked features, which should more or less the same for both methods since for SVMRFE these are determined on the earliest iterations of the algorithm, being thus closely in behavior to the single run of SVMONE. Moving to lower cardinalities the instability of SVMRFE increases due to the already mentioned fact that small differences in the coefficients can inverse the rank and thus remove different features. The difference in instability between SVMONE and SVMRFE increases as we move to lower cardinalities beacause there the final feature sets of SVMRFE are determined in the last iterations of the support vector machine algorithm.

5 Stability and Classification Performance

A feature selection algorithm, (FSA for brevity), alone can provide an indication of which features are informative for classification but it cannot provide an estimate of the discriminatory information of these features, since it does not construct classification models whose error could be estimated. In the same manner stability results cannot provide the sole basis on which to select an appropriate FSA; nevertheless they can support the selection of an FSA when the latter is coupled with a classification algorithm, (CA for brevity), and enhance the confidence of the users on the analysis results (provided that the FSA is found to be stable). The final selection can be based on a combined evaluation of stability and classification performance.

The simplest scenario goes as follows, couple a given CA with a number of FSAs and estimate the classification performance and the stability of the FSA using the process

of the discretization process that discretized the corresponding features to a single bin.

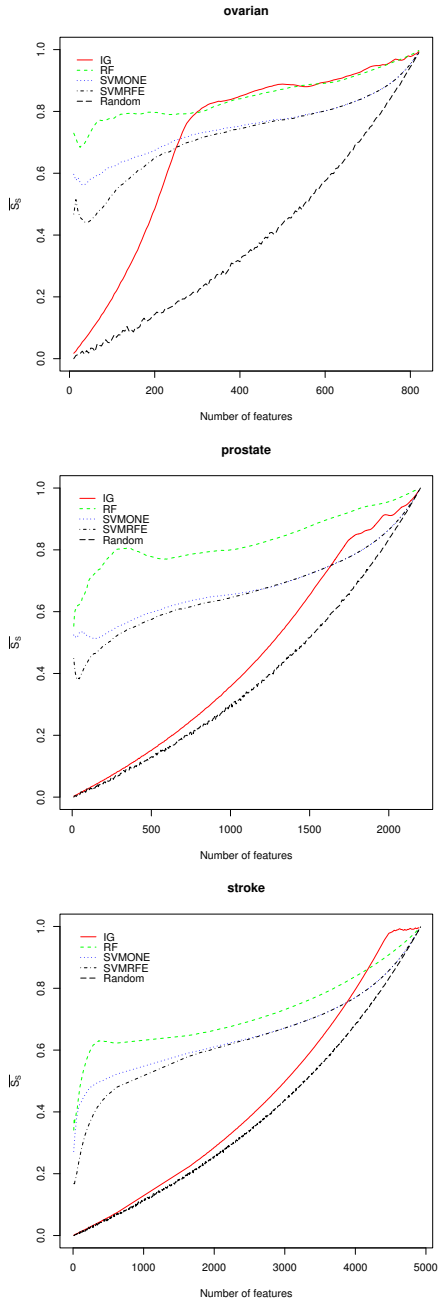

Figure 2. $\overline{S_S}$ plots for varying cardinalities of the final feature set.

described earlier. Then calculate the statistical significance of error differences. Among the combinations of the CA and FSAs that were found to be better than all the others choose the combination that contains the most stable FSA.

To demonstrate the above idea we selected as classification algorithm the linear SVM with the complexity parameter set to 0.5. We performed a series of experiments in which each FSA was paired with the CA. In each experiment we fixed the number of selected features to N. We ranged N from ten to 50 with a step of ten. For a given N the four pairs of FSA-CA were compared with respect to their classification error and the stability of the FSA. Statistical significance of error differences is computed by McNemar's test of significance (sig. level=0.05). The complete results are given in table 3. Each row of that table gives the classification errors of a FSA-CA pair followed by the stability estimate, $\overline{S_S}$, of the FSA. The errors of the FSA-CA pairs that get the top positions, for a given N, without being significantly different between them are typed in *italics*.

Applying the selection scenario mentioned above we see that for stroke and ovarian and for different values of N there are several FSA-CA pairs that are indistinguishable in terms of classification error. Consider the stroke dataset with $N = 10$; Information Gain, ReliefF and SVMRFE have similar classification performance. In this case we can also consider their stability. ReliefF is by far the most stable with an $\overline{S_S}$ value that is double of that of SVMRFE and more than an order of magnitude greater than that of Information Gain. Obviously the advantage of selecting the most stable FSA is that we have much more confidence on the features. Moreover coupling the results with a visual representation of stability as the one given in figure 1 provides a clear picture of the important features and how robust they are to perturbations of the training set.

One question that arises from the above results is: how is it possible for a FSA to be very unstable and still when coupled with a CA to produce good results. This was actually the case many times with SVMRFE. For example in the stroke dataset and $N = 20$ SVMRFE coupled with the CA was signigicantly better than the other three FSA-SA paits. Nevertheless its $\overline{S_S}$ estimate was 0.16 (in feature sets of cardinality 20 this corresponds to an average of 5.5 common features). One possible answer to that is redundancy. Among the initial full feature set there are possibly many different subsets of cardinality 20 on which classification models can be constructed that can accurately predict the target concept[3]. Cases like that, i.e., instability coupled with high classification performance, can be simply an indication of redundancy within the full feature set. This also means that the feature selection algorithm under examination does not have a robust way to tackle redundancy.

[3]This is true for the mass-spectrometry applications due to the nature of the preprocessing techniques that are applied on them

6 Conclusions and Future Work

To the best of our knowledge this is the first time that a framework that measures the stability of feature selection algorithms is proposed. We defined the stability of feature selection algorithms as the sensitivity of the "feature preferences" that they produce to training set perturbations. We examined three different stability measures and proposed a resampling technique to empirically estimate them. The most interesting one was based on S_S a measure of the overlap of two feature sets. We exploited the framework to investigate the stability of some well known feature selection algorithms on three datasets coming from the domain of proteomics and gained some interesting insights. Stability can be also used to support the selection of a feature selection algorithm.

We believe that the notion of stability is central in real world application where the goal is to determine the most important features. If these features are consistent among models created from different traning data the confidence of the users on the analysis results highly increases. The results of the empirical estimation of stability can be elegantly visualized and provide a clear picture of the relevant features, their robustnes to different training sets, and the stability of the feature selection algorithm.

Future work includes the examination of stability of more algorithms on a bigger and more diverse set of problems; refining the S_R stability measure in order to reflect better large differences on the top ranked features; aggregating the different feature sets produced from subsamples of a given training set in what can be viewed as the analogue of ensemble learning and model combination for feature selection; finally we would like to examine the possibility of using the stability profiles to select the appropriate number of features (the knots in the stability graphs).

Stroke				
N	IG	Relief	SVM	SVMRFE
10	*32.22*-0.02	*30.29*-0.34	37.02-0.27	*26.45*-0.16
20	28.85-0.05	28.85-0.36	35.10-0.30	*21.64*-0.16
30	*27.89*-0.09	*27.41*-0.37	*28.37*-0.33	*23.56*-0.17
40	*29.81*-0.12	*25.97*-0.38	*25.00*-0.35	*25.49*-0.18
50	*27.89*-0.16	*28.37*-0.40	*26.45*-0.37	*25.49*-0.19
Ovarian				
N	IG	Relief	SVM	SVMRFE
10	10.28-0.01	10.28-0.72	07.12-0.59	*01.19*-0.46
20	05.56-0.06	05.93-0.69	*03.96*-0.58	*01.19*-0.47
30	04.75-0.09	*01.59*-0.69	*01.19*-0.56	*00.40*-0.45
40	03.17-0.12	*01.59*-0.69	*00.40*-0.56	*00.40*-0.44
50	*02.77*-0.16	*01.59*-0.70	*00.40*-0.58	*00.40*-0.44
Prostate				
N	IG	Relief	SVM	SVMRFE
10	18.64-0.01	18.95-0.55	18.02-0.52	*13.05*-0.44
20	17.71-0.01	17.09-0.60	16.46-0.51	*11.50*-0.40
30	16.46-0.02	15.84-0.61	14.91-0.52	*10.87*-0.38
40	16.15-0.03	14.91-0.62	13.36-0.52	*09.01*-0.38
50	14.60-0.04	13.36-0.62	13.05-0.53	*09.32*-0.38

Table 3. Classification error estimations coupled with $\overline{S_S}$ stability estimation of the feature selection method. N is the number of selected features.

References

[1] P. Domingos. A unified bias-variance decomposition and its applications. In P. Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 231–238. Morgan Kaufmann, 2000.

[2] R. Duda, P. Hart, and D. Stork. *Pattern Classification and Scene Analysis*. John Willey and Sons, 2001.

[3] U. Fayyad and K. Irani. Multi–interval discretization of continuous attributes as preprocessing for classification learning. In R. Bajcsy, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993.

[4] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

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

[6] A. Kalousis, J. Prados, E. Rexhepaj, and M. Hilario. Feature extraction from mass spectra for classification. 2005. Submitted to 6th European Conference on Principles and Practice of Knowledge Discovery in Databases.

[7] E. Petricoin, A. Ardekani, B. Hitt, P. Levine, V. Fusaro, S. Steinberg, G. Mills, C. Simone, D. Fishman, E. Kohn, and L. Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 395:572–577, 2002.

[8] E. Petricoin, D. Ornstein, C. Paweletz, A. Ardekani, P. Hackett, B. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. Simone, P. Levine, W. Marston Linehan, M. Emmert-Buck, S. Steinberg, E. Kohn, and L. Liotta. Serum proteomic patterns for detection of prostate cancer. *Journal of the NCI*, 94(20), 2002.

[9] J. Prados, A. Kalousis, J.-C. Sanchez, L. Allard, O. Carrette, and M. Hilario. Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. *Proteomics*, 4(8):2320–2332, 2004.

[10] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1–2):23–69, 2003.

[11] P. Turney. Technical note: Bias and the quantification of stability. *Machine Learning*, 20:23–33, 1995.

[12] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.