Bias and Stability of Single Variable Classifiers for Feature Ranking and Selection

Shobeir Fakhraei[a,b], Hamid Soltanian-Zadeh[a,c], Farshad Fotouhi[d]

[a]*Medical Image Analysis Laboratory, Department of Radiology, Henry Ford Health System, Detroit, MI 48202, USA*
[b]*Department of Computer Science, University of Maryland, College Park, MD 20740, USA*
[c]*Control and Intelligent Processing Center of Excellence (CIPCE), School of Electrical and Computer Engineering, University of Tehran, Tehran 14395, Iran*
[d]*College of Engineering, Wayne State University, Detroit, MI 48202, USA*

Abstract

Feature rankings are often used for supervised dimension reduction especially when discriminating power of each feature is of interest, dimensionality of dataset is extremely high, or computational power is limited to perform more complicated methods. In practice, it is recommended to start dimension reduction via simple methods such as feature rankings before applying more complex approaches. Single Variable Classifier (SVC) ranking is a feature ranking based on the predictive performance of a classifier built using only a single feature. While benefiting from capabilities of classifiers, this ranking method is not as computationally intensive as wrappers. In this paper, we report the results of an extensive study on the bias and stability of such feature ranking method. We study whether the classifiers influence the SVC rankings or the discriminative power of features themselves has a dominant impact on the final rankings. We show the common intuition of using the same classifier for feature ranking and final classification does not always result in the best prediction performance. We then study if heterogeneous classifiers ensemble approaches provide more unbiased rankings and if they improve final classification performance. Furthermore, we calculate an empirical prediction performance loss for using the same classifier in SVC feature ranking and final classification from the optimal choices.

Keywords: Feature Ranking, Feature Selection, Bias, Stability, Single Variable Classifier, Dimension Reduction, Support Vector Machines, Naïve Bayes, Multilayer Perceptron, K-Nearest Neighbors, Logistic Regression, AdaBoost, Random Forests

1. Introduction and Motivation

Due to the use of cross validation, classifier-based feature selection/ranking methods often demonstrate superior performance comparing to methods that only use training error (Guyon, 2008) . Wrapper feature selection (Kohavi and John, 1997) and Single Variable Classifier (SVC) feature ranking methods (Guyon and Elisseeff, 2003) are two common methods of utilizing classifiers in this context. While multiple aspects of wrappers have been studied, SVCs have not received much attention in the scientific literature. In this paper, we study the stability and bias of SVC feature ranking methods and report problems that affect both SVC and wrapper methods in practice.

When using a classifier to rank features, it is interesting to find out whether that the result is a relatively universal or the classifier's influence on the result makes it specific to that classifier. We have conducted multiple experiments to study such questions. Furthermore, it is interesting to found out if the rankings are highly affected by the classifiers, which classifiers generate more similar results. This information provides insight on which classifiers could use rankings interchangeably, and also provides interesting information for generating ensemble rankings using methods with less correlated results. We empirically study and report this similarity for multiple classifiers.

It is also a common intuition that a better optimized feature selection/ranking is achieved when the same classifier is used in the selection/ranking process as an evaluation criterion. We empirically study this and report interesting results that are somewhat contradictory with this intuition. In other words, we study how better would the final classification performance be if we use

the same classifier for feature ranking and final classification, compared to situations in which different classifiers are used in the two steps.

Superiority of ensemble methods in multiple domains raises the question about the possibility of finding a universally superior ranking method based on ensemble of multiple rankings. This question would be of high interest especially if the optimal ranking for a classification task could not be achieved via using the same classifier's performance as the ranking criteria. We study the performance of an ensemble ranking and compare it to other SVCs.

Finally, we investigate that if using the same classifier for feature ranking and final classification is not optimal, how far it is from the best empirically found approach. In other worlds, is the extra effort of finding a better ranking worth it? Or using the same classifier for ranking and final classification is good enough?

In the rest of the paper, first an overall introduction to dimension reduction and feature selection and ranking is provided. Then, some information about SVC feature rankings is presented, and bias and stability of such feature ranking methods are formally defined. The five questions we address in the study are defined and framework and study method are presented. Next, empirical results based on multiple dataset are reported. Finally, insights into the reasons behind the observed phenomena are provided and some practical guidelines to gain superior results are described.

1.1. Dimension Reduction

To fully benefit from the power of data mining algorithms and learning models, data have to be prepared to fit the requirements of the desired learning model. Such preparation which is commonly referred to as data pre-processing might contain tasks such as aggregation, sampling, discretization, data cleansing, normalization, and dimension reduction. Dimension reduction which is the process of reducing the number of variables[1] under consideration, is a significantly important step which has received considerable attention from the research community (Huan and Lei, 2005). It is well known that the prediction performance of learning models degrades when faced with many variables that are not necessary for predicting the desired output (Kohavi and John, 1997). Curse of dimensionality is perhaps the most well known phenomena that happens when the number of variables increases dramatically (Bishop, 2006, Hastie et al., 2009).

[1] We use *variable*, *feature* and *attribute* terms interchangeably.

There are many reasons that collected data might contain unwanted variables, thus making dimension reduction a necessity. Often due to convenience and low cost, data collection is overdone. Most of the time, data is not even collected for the purpose of data mining, thus including many unrelated variables. In occasions, it is not known which variable is most appropriate for learning beforehand, therefore, all presumably related variables are recorded.

For such reasons, dimension reduction has become an essential step in application of data mining and machine learning in many domains like text mining, image retrieval, microarray data analysis, protein classification, face recognition, handwriting recognition, intrusion detection, and biomedical data analysis. While dimension reduction is as old as machine learning itself, exponential increases in the number of variables in recent years have made dimension reduction significantly important in several domains (Guyon and Elisseeff, 2003, Zhao and Liu, 2011).

In some of these domains, the number of variables even exceeds the number of samples. Such scenarios would result in poor performance due to over-fitting and make dimension reduction inevitable (Hastie et al., 2009). For example, according to Probably Approximately Correct (PAC) learning theory, a dataset with a binary class and n binary variables has a hypothesis space of size 2^{2^n}, and it would require $O(2^n)$ samples to learn a PAC hypothesis without any inductive bias (Loscalzo et al., 2009). Therefore, with an increase in the number of variables (n), collection of more samples is necessary to avoid over-fitting, which is not always possible.

Dimension reduction methods may be applied to a dataset with several objectives. It may be used to eliminate unrelated features and improve the performance of the model. It may also decrease learning time, measurement efforts, and storage space. Other important aspect of dimension reduction is the acquisition of knowledge about the data and features themselves. Finding which gene is more discriminative of a condition, or which medical test is more informative with regards to a diagnosis is a valuable discovery. Finally, dimension reduction may be used to achieve the best reconstruction of the data with a minimum number of variables; such objective however is less relevant to machine learning and is more useful for data compression.

In a typical machine learning and data mining classification problem a dataset (DS) is provided via a collection of m-dimensional vectors (x^j) or data points where each element of the vector corresponds to a value from a feature (f_i). A class label (y^j) is also provided for each

vector that maps each data point to a particular class. Dataset is formally defined in $\mathbb{R}^n * \mathbb{R}^{(m+1)}$ space with (X, F, Y) where X contains n instances of x^j such that $x^j = (f_1^j, \ldots, f_m^j) \in \mathbb{R}^m$ and $j = 1, \ldots, n$. Each feature (f_i) comes from collection of features F and f_i^j indicates the value of that feature for a particular instance. A classifier (C_L) optimizes a criteria (λ) to be able to predict the class labels of unobserved instances as correctly as possible. Example of such criteria could be mean square error, accuracy or area under the ROC curve which is discussed in later sections. To facilitate the argument and without loss of generality, we assume that λ should always be maximal. Dimension reduction aims to maximally preserve the useful information in the original data according to some criterion and discard the unnecessary information. In other words, dimension reduction maps DS from (X, F, Y) to (X', F', Y) where X' contains vectors such as $x'^j = (f_1'^j, \ldots, f_{m'}'^j) \in \mathbb{R}^{m'}$ and $m' < m$ and $\lambda(X', F', Y) \geq \lambda(X, F, Y) - \epsilon$ for some criterion λ and some estimation parameter ϵ which we will not include in the rest of our discussions for simplicity.

1.2. Feature Selection and Ranking

Dimension reduction may be broadly categorized into two main groups; *feature extraction/construction* and *feature selection*. In feature extraction, new features are constructed based on the original features in the datasets. Principal Component Analysis (PCA), Isometric Feature Mapping (IsoMap) and Locally Linear Embedding (LLE) are examples of such approach (Lee and Verleysen, 2007). In feature extraction the following characteristics hold for newly generated feature:

$$(f_i' \in F') \wedge (f_i' \notin F) \wedge \left(f_i' = \varphi(f_j, \ldots, f_k); f_i \in F\right)$$

Although *feature extraction* methods are used in many domains to improve performance of the learning models, they suffer from limitations with respect to other objectives of dimension reduction. Since the original features are not the output of these processes, knowledge about the discriminative value of each feature is not acquired easily. On the other hand, most of these methods construct new features based on all original features and do not reduce the measurement requirements of the data gathering.

Feature selection on the other hand is the process of choosing or prioritizing a subset of original features based on a criterion. In general the following characteristics hold for features in the new feature space:

$$(f_i' \in F') \wedge (f_i' \in F) \wedge (|F'| < |F|)$$

where $|F^*|$ indicates number of features in F^*.
Feature selection in general can itself be divided into several subgroups from different perspectives (Guyon and Elisseeff, 2003, Huan and Lei, 2005, Kira and Rendell, 1992), one of which is the output of the feature selection process that can be categorized as the following:

- **Feature subset selection:** Here, the output is an optimal feature subset that is aimed to maximize performance of the model. The features in the selected subset are not prioritized against each other. Should the subset size be reduced, there is no information regarding which features to eliminate first from the group. In this group, a feature subset F' is provided such that:

$$F' = \{f_1', \ldots, f_i'\} \; and$$
$$\lambda(X', F', Y) \geq \lambda(X, F, Y)$$

- **Nested subsets of features:** As the name suggests, the output of this group of algorithms is a list of features that form a nested structure of feature subsets. There could be two types of nested feature subsets depending on the search strategy; *backward* and *forward* selection. As an example, when forward selected, a nested list of feature subsets $R = \langle f_1', f_2', f_3', \ldots \rangle$ is returned by an algorithm. This means that best subset containing only a single feature is f_1' and best subset of two features with the condition of already having f_1' in the subset is $\left\{f_1', f_2'\right\}$. However, this list does not indicate individual preference of f_2' over f_3', but it states that the former demonstrates superior performance in combination with f_1' comparing to the later. A backward elimination nested feature subset conveys the same logic with a reverse order. In this group, a list of feature subsets R is provided such that:

$$R = \langle f_1', \ldots, f_i' \rangle \; ; \; \forall \left(f_i', f_j' \in R\right) : i < j \rightarrow$$
$$\lambda\left(X', R_{i-1} \cup \{f_i'\}, Y\right) \geq \lambda\left(X', R_{i-1} \cup \{f_j'\}, Y\right)$$

where

$$R_k = \{f_1', \ldots, f_k'\}$$

- **Feature ranking:** This group of algorithms sort the features based on a quality index that reflects the individual relevance, information, or discriminative capacity of a feature. Therefore, when a list

such as $R = \langle f'_1, f'_2, f'_3, \dots \rangle$ is produced in the feature ranking fashion, it suggests that f'_2 is superior to f'_3 individually with respect to the quality index. In this group, a list of feature subsets R is provided such that:

$$R = \langle f'_1, \dots, f'_i \rangle \; ; \; \forall (f'_i, f'_j \in R) : \\ i < j \rightarrow \lambda(X', \{f'_i\}, Y) \geq \lambda(X', \{f'_j\}, Y)$$

A portion of the top features in the later two approaches may be selected as a feature subset to build a classifier in several ways (Ruiz et al., 2006). User mandated constrains such as the number of desired features or performance threshold, and use of randomly generated features as a probe to discriminate between related and unrelated features such as the method summarized by Stoppiglia et al. (2003) are common approaches for this purpose. Plotting a learning or an error curve, constructed by adding a feature at a time to the subset and evaluating the subset with a classifier is another way of finding the best portion of the highly ranked features for subset construction (Slavkov et al., 2010). With respect to the statistical methods summarized by Demsar (2006), different cut-off criteria have been studied by Arauzo-Azofra et al. (2011).

The final objective of feature selection determines which one of the above categories is preferable. When maximizing the performance of a learning model is the only goal, feature subset selection algorithms would be sufficient. Nested subsets of features are useful when information about the features values with consideration to their interaction is desirable and user demands more insight and supervision in the final selected subset. However, since feature selection maps to a search problem in a lattice (Kohavi and John, 1997), most greedy hill-climbing algorithms generate a nested subset of features where the latest generated subset is the outcome.

Feature ranking is mostly desirable when knowledge about the discriminative value of individual features is of interest. For example, in medicine or bioinformatics, when each feature corresponds to a medical test, biometric, or a gene, the result of a feature ranking algorithm itself is of great value.

The feature selection algorithms that evaluate subset of features are called *multivariate* feature selection methods. Feature ranking on the other hand usually evaluates performance of each feature individually, and is categorized as a univariate method. When interactions between features are important, *univariate* methods fail to capture such phenomena of interest and multivariate methods must be used (Guyon and Elisseeff, 2003). It should also be noted that although multivariate methods lead to more universal predictors than univariate methods, they may result in less predictive performances due to over-fitting (Guyon et al., 2005). Furthermore, due to the time consuming nature of the search algorithms used in feature selection, feature ranking methods are widely used in practice.

Based on the use of a learning model, feature selection algorithms are divided into *wrapper*, *filter*, *embedded* and *hybrid* methods. The filter methods evaluate feature subsets based on statistical measures, while the wrapper methods use a learning model such as a classifier for that matter. Embedded approaches also use a learning model but instead of features predictive or discriminative performance, internal factors of the learning model are considered as the evaluation criteria; SVM-RFE is an example in this category (Guyon et al., 2002). Hybrid methods generate feature subsets based on statistical measures and evaluate the best selected subset using a learning model to benefit from both filter and wrapper approaches (Peng et al., 2010, Bacauskiene et al., 2009, Gheyas and Smith, 2010). Due to use of *training error* in statistical tests, cross-validation based feature selection/ranking methods often demonstrate superior results compared to statistical tests.

Another group of algorithms categorized as penalty-based methods tend to address this problem via augmenting the *training error* by a penalty term (Guyon, 2008). A successful algorithm in this category which has generated major interest is the *least absolute shrinkage and selection operator (LASSO)* (Hastie et al., 2009, Tsanas et al., 2010, Tibshirani, 1996) . This is a penalized least squares method imposing an ℓ_1-norm on the regression coefficients. It does both continuous shrinkage and automatic variable selection simultaneously. The ℓ_1-norm promotes sparsity (some coefficients become zero), and therefore the LASSO can be used as a feature selection method. Use of elastic nets is another method in this group which reportedly outperform LASSO when the number of features is highly greater than the number of samples (Zou and Hastie, 2005).

While filter methods are faster, they are shown to be less accurate than wrapper approaches, which on the other hand, have better accuracy but are computationally expensive and are likely to over-fit the selected feature subset to a specific learning model (Chrysostomou et al., 2008). The classifier bias in the wrapper methods promotes the filter methods to be stronger candidates in cases where features independent discriminative powers are of interest, such as in feature ranking. In such cases, regardless of the learning model, the user is interested in knowing which feature is superior to the others with respect to a criterion such as class discrimination.

Nevertheless, classifiers are being used in feature ranking. Single Variable Classifier (SVC) is a feature ranking method where variables are ranked according to their individual predictive power. In other words, features are ranked based on the performance of a classifier (C_R) built with just that single variable (Guyon and Elisseeff, 2003). While benefiting from the accuracy of classifiers, this method of ranking is not as computationally intensive as a full wrapper. In terms of classification performance this method has been shown to be superior to filters (Fakhraei et al., 2010a). For clarity we denote the classifiers that are used for ranking the features with C_R and the classifiers that are used for the final classification (learning) with C_L.

2. Bias and Stability of SVCs

Intuitively, the classifier (C_R) used in SVCs for feature ranking affects the ranking results. It is reasonable to think SVC feature ranking is biased towards a classifier (C_R), because features that satisfy the characteristics of such classifier would be ranked higher than their real discriminative power. As an example from two features that completely separate the instances in each class, the feature that separates the instances of classes in a linear fashion receives a higher score when evaluated with a linear classifier.

A visual example of such bias is demonstrated in Figure 1 for one feature. While both features have the same capability to completely discriminate the classes, when evaluated with cross validation, the feature in Figure 1.(a) should be favored when a linear classifier (e.g., linear Support Vector Machine (SVM)) is used. On the other hand, the feature in Figure 1.(b) should be given a higher rank when a nonlinear classifier such as an instance based classifier like K-Nearest Neighbors (KNN) is used.

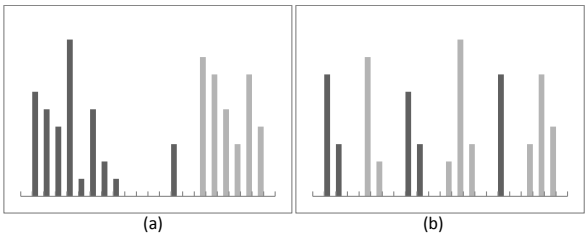

Figure 1: Histogram of features with linear and nonlinear class discrimination. The feature shown in (a) that linearly separates the classes should be favored with a linear classifier as more discriminative while the feature in (b) that separates the classes nonlinearly should be favored with an instance based classifier such a KNN.

To formally define such bias, let's assume that a ranking list R^{C_R} is provided using SVC feature ranking with classifier C_R and $R^{C_R}_\alpha$ indicates a feature subset that includes the best features of that list, and λ_{C_L} indicates the classification performance criteria (e.g., accuracy or AUC) evaluated via using the classifier C_L. A feature ranking demonstrates bias towards a certain classifier when:

$$R^{C_R} = \langle f'_1, \ldots, f'_i \rangle;$$
$$\forall C_R \neq C_L : \lambda_{C_L}(X', R^{C_L}_\alpha, Y) \geq \lambda_{C_L}(X', R^{C_R}_\alpha, Y)$$

where $R^{C_R}_\alpha = \{f'_1, \ldots, f'_\alpha\}$.
This definition of bias depends on the value of α which could be defined in various ways that are explained in next sections. However, as it would be clarified in more details in later sections, to acquire a more general conclusion about such bias, we use multiple values for α in our experiments and report the average result.

There is also the issue of stability of feature selection and ranking methods especially in high dimensional datasets with small sample sizes. A feature selection algorithm may select largely different subsets of features, under variations to the training data. These different feature subsets could even generate relatively close classification performance. In such cases, obtaining knowledge about the features themselves is not easy. *Stability* or *robustness* of feature selection methods with respect to variations in the training data is a topic of recent interest among researchers (Gulgezen et al., 2009, Kalousis et al., 2007, Yu et al., 2008, Saeys et al., 2008). In this paper, we have studied the stability of SVC feature rankings with regards to change of classifiers.

In general, stability of a feature selection method is measured as the effect of the method's internal conditions variation on the variation of the final selection results. In other words, stability of feature selection methods are measured in terms of the similarity of their result under variations of conditions. In our scenario, if ω indicates similarity (e.g., correlation) of two sets, and two different rankings $R^{C^1_R}$ and $R^{C^2_R}$ were generated using different classifiers in SVC, then the similarity of the two rankings would be defined as:

$$\omega(R^{C^1_R}, R^{C^2_R})$$

where this similarity could be reported as a number in $[0, 1]$ range or as a percentage. Higher values of this number are indications of the feature selection method's higher stability with variations in the ranking classifier (C_R).

In this paper, we have shown how instable the SVC feature ranking results could be using seven classifiers and eleven datasets, and studied the bias of classifiers in SVC ranking. We also measured the similarity and correlation of the results obtained using different classifiers and studied if using a heterogonous ensemble of the results from several classifiers reduces the bias.

3. Related Work

In recent years, stability of feature selection methods has been studied with respect to variations in the training set. Stability of a feature selection algorithm is usually defined as the robustness of the feature preferences produced relative to the differences in the training sets drawn from the same generating distribution (Kalousis et al., 2007). Zou (2006) has studied the stability of LASSO and necessary conditions for its consistency and has proposed adaptive LASSO, where adaptive weights are used for penalizing different coefficients in the ℓ_1-norm.

Several stability measures have been studied and proposed to measure the similarity of feature selection via variation in the datasets. Novovicová et al. (2009) proposed a new stability measure based on the Shannon entropy to evaluate the overall occurrence of individual features in selected subsets of possibly varying cardinality. Kalousis et al. (2005) studied several similarity measures to quantify the stability of feature preferences via performing a series of experiments with several feature selection algorithms on a set of proteomics datasets. Yu et al. (2008) and Loscalzo et al. (2009) have proposed stable feature selection methods that cluster features and choose representatives of each cluster for the final feature subset.

Using this definition, stability is measured with respect to how similar the feature selection results are, using multiple instance samples drawn from a dataset. Our study, however, instead of changing the sample population, changes the classifier used in the SVC feature ranking and studies the effect of this change on the stability of the final selected features.

Feature selection bias has also been studied in recent years. Choosing features based on the correlation with the class variable may introduce an optimistic bias, in which the response variable appears to be more predictable than it actually is. This is because the high correlation of the selected features with the response may be due to chance (Li et al., 2008). With respect to this issue a Bayesian method for making well-calibrated predictions has been proposed by Li et al. (2008).

In another study, Singhi and Liu (2006) have stated that using the same training dataset in both feature selection and learning can result in so called feature subset selection bias. Motivated by the research done on the selection bias in regression, statistical properties of feature selection bias in classification are analyzed. They have shown how this bias impacts classification learning via various experiments, and shown the selection bias has less negative effect in classification than in regression due to the disparate functions of the two.

In this paper, however, we study the bias of each classifier with respect to the final classification performance. In other words, we study how better would the classification performance be if we use the same classifier for feature ranking and final classification, compared to situations in which different classifiers are used in each step.

There are also multiple experimental studies that compare performances in feature ranking and feature selection methods. As an example, Hall and Holmes (2003) provided a benchmark for comparison of feature selection methods that work based on a ranked list of features. Naïve Bayes and decision trees classification algorithms have been used for the evaluation of the feature selection methods. In another study, nine common performance metrics are compared in a wrapper based feature ranking method described by Altidor et al. (2009) and their correlations have been reported. They have ranked the features based on cross validation with SVC and calculation of *Feature Risk Impact*. This study identifies metrics that provide relatively same feature rankings. It is also demonstrated that area under *Precision-Recall curve (APRC)* and area under *Receiver Operating Characteristic curve (AUC)* are somewhat correlated in most of the domains and classifiers. In our studies, we report similarities and correlations of the rankings when using different classifiers in the SVC rankings.

Feature ranking and feature selection ensembles have also been studied from different perspectives in recent years and their benefits are reported in several publications. In an effort to address the non-robust behavior of feature rankings in different datasets and different classifiers, Makrehchi and Kamel (2007) combine eight different ranking metrics with different combination functions. Similarly, Yan (2007) have studied fusion of multiple criteria for feature ranking and combined several ranking measures into a united rank.

In a credit scoring application, Chen and Li (2010) report results of a study on relative effectiveness of feature selection methods based on *Linear Discriminate Analysis (LDA)*, *Rough Sets Theory (RST)*, *Decision Tree*

(DT), *F-score* and their combination when the classification is performed via SVM classifiers. Bryll et al. (2003) have proposed an ensemble method based on attribute bagging and have studied the stability of the classifiers participating in the ensemble by considering the standard deviation of their accuracies. Fakhraei et al. (2010a) have shown that a consensus feature ranking based on SVCs outperform *Chi-Square* and *Information Gain* in a dataset with missing values, have studied and positive and negative effects of classifiers in the consensus ranking (Fakhraei et al., 2010b).

Santana and Canuto (2014) identified that the wrapper methods are strongly coupled to the classification algorithm, having to run again when switching classification methods. They proposed to use different hyperparameter setting, training dataset and classifier types to enhance the diversity of an ensemble method. In an ensemble setting via using particle swarm, ant-colony and genetic algorithms optimization techniques they chose subsets of features for the individual components of ensembles to enhance the performance. Cruz et al. (2013) proposed use of ensemble methods based on multiple feature representations. They used a dissimilarity and the intersection of errors, to analyze the relationships among feature representations which they use to train classifiers. They showed efficiency of this approach to handwritten character and digit recognition.

Cho (2014) used genetic algorithms to search the space of single variable classifiers. They combined the result of several selected SVCs to generate the final classification and showed enhanced performance on handwritten digit recognition. You et al. (2014) proposed a feature ranking method based on partial least squares and feature elimination and applied it to multiclass classification problems. Alt (2013) proposed single variable classifiers for feature extraction in binary text categorization, by using the outputs of the SVCs to form the document vectors. Yoon et al. (2013) proposed a method to integrate the feature selection and classification for neural networks. Zhao et al. (2011) proposed a method based on genetic algorithm to simultaneously optimize the feature subset and the parameters for SVM.

4. Proposed Study

We have investigated the following five questions related to application of single variable classifiers in feature ranking:

1. Does the classifier used in SVC ranking have a huge impact on final feature ranking results or the discriminative power of features themselves is more important than the classifier?
2. Which classifiers produce more similar results when used in SVC feature ranking?
3. If choice of classifiers influences the final rankings, is there a classifier bias in the rankings? In other words, when using the final feature ranking result to build a model for classification, is it always best to use the same classifier for feature ranking and final classification or other combinations work better?
4. Does taking an ensemble approach and combining the results from SVC feature rankings via multiple classifiers help getting a universally superior result that improves classification performance?
5. If using the same classifier for feature ranking and final classification is not optimal, what is the performance loss of taking such approach comparing to the optimal choice of classifiers?

We have carried out multiple experiments to answer the above questions. To answer the first question, we have compared the results from SVC rankings using different classifiers. We reported the correlation and similarity of the rankings, to address the second question. For the third, we have evaluated a top portion of ranked features using different classifiers and reported their comparative performance results. With respect to the forth question, we have calculated ensemble scores and rankings and compared their performances with single classifier SVC rankings. Finally, we have calculated the classification performance difference between the best combination of ranking classifiers (C_R) and learning classifiers (C_L) and the where the same classifier is used for both of them, to answer the fifth question.

In the following section, the overall framework of the experiments has been discussed and ranking and similarity measures that we have used in the experiments are explained. Calculations of ensemble ranking and methods for classification performance evaluation are also included in this section.

4.1. Framework

The overall framework of the experiments is shown in Figure 2. At the first step, single variable classifiers are built using each of the classifiers (C_R) included in the study and using each of the features in the datasets. Then, the features are ranked based on the classification performance of the SVC (C_R) built with that feature. Correlation of the ranking with different classifiers (C_R) and their intersection of results are measured in this

step as their similarity (ω) for each dataset and the overall averages reported. An ensemble performance score is also assigned to each feature by combining the results from different classifiers (C_R). This ensemble rank will be used to answer the fourth question asked in the previous section. In other stages, portion of the top ranking features from each ranking list has been selected into a feature subset and several classifiers (C_L) have been built using these features. Via 5-fold cross validation, performances of the classifiers (C_L) have been recorded. These experiments have been repeated for different values of and the mean and the standard deviation of the results reported.

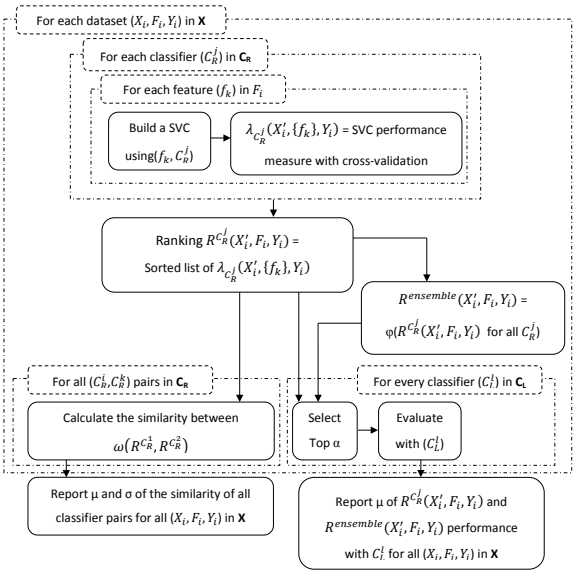

Figure 2: Experimental schema of the study and evaluation methods. In the experiments, for each dataset, correlations and similarities of the SVCs built using different classifiers (C_R) are calculated and the final correlation between the SVCs reported as the average correlation on different datasets. In other parts of the study, feature subsets are selected from the top of the feature ranking lists and these subsets are used for classification to measure which subset works best with which classifier (C_L), and answer the related question. The reported results are the average numbers over all datasets.

4.2. Ranking Measure

Several criteria may be used to measure the classification performance of a SVC, such as accuracy, precision or positive predictive value (ppv), negative predictive value (npv), recall or sensitivity or true positive rate (tpr), specificity or true negative rate (tnr), false positive rate (fpr), false negative rate (fnr), and F_1-measure. Definitions of these measures could be found in (Forman, 2003) as well as most classic data mining and machine learning books and resources. However, most of these measures calculate the model performance based on a fixed decision boundary which causes problems with class imbalance datasets.

Since datasets with class imbalance distributions are common in real world applications, performance of many classifiers are evaluated by the receiver operating characteristic (ROC) curve which is constructed by plotting the true positive rate versus the false positive rate by changing the decision threshold or boundary (Fawcett, 2004). Area under the ROC curve (AUC) is a performance measure that can be used to evaluate the SVC, despite the imbalanced distribution of the classes. Methods of Chen and Wasikowski (2008) and Wang and Tang (2009) that use AUC for feature selection are examples in this category. We have used AUC in our studies to evaluate the classification performance.

4.3. Feature Subset Selection

Different thresholds may be used to select feature subsets from the top feature ranking lists. Selecting a fixed number of features, a fixed percentage of the features, using a performance threshold and using the amount of performance improvement when adding a new feature to the subset, can be used for this purpose (Arauzo-Azofra et al., 2011). Since we mostly want to use datasets with very high dimensionality, using a method that works based on measuring the performance improvement of adding features is not feasible. On the other hand, since the numbers of features in the datasets are very different, using a fixed percentage of the features is not practical either. For example, using 10% threshold in datasets with 20,000 feature and 44 features produce incomparable results. Hence, for the sake of consistency across datasets, we have chosen a fixed number of features in the subsets, with respect to being computationally feasible and relevant to the datasets average dimensionality.

4.4. Similarity Measure

In the study of stability of feature selection methods, different similarity measures have been proposed and utilized (Kalousis et al., 2007, Yu et al., 2008, Saeys et al., 2008). In our studies, we have considered two perspectives. One is how the whole ranking list generated by one classifier is similar to the rankings generated by the other. This similarity (ω) is measured by the Spearman's rank correlation coefficient, which for a ranking using classifiers C_R^1 and C_R^2 is defined as:

$$\omega_\rho\left(R^{C_R^1}, R^{C_R^2}\right) = 1 - \frac{6\sum_{i=1}^{m}\left(R^{C_R^1}(f'_i) - R^{C_R^2}(f'_i)\right)^2}{m(m^2-1)}$$

where m is the total number of features in a dataset and $R^{C^1_R}(f'_i)$ and $R^{C^2_R}(f'_i)$ are the rank of feature f'_i with SVCs built with classifiers C^1_R and C^2_R.

Besides SVCs, classifiers are often used in wrapper feature selection algorithms where a feature or a group of features with the best or the worst performance are selected for the next step of the iteration, to either be added to or eliminated from the final subset. With this practice in mind, we have also reported the intersections of the features in the top and bottom portion of the ranked list obtained using different classifiers (C_R). Intersection of the results generated using classifiers C^1_R and C^2_R is reported as:

$$\omega_\cap\left(R^{C^1_R}, R^{C^2_R}\right) = \frac{\left|R^{C^1_R}_\alpha \cap R^{C^2_R}_\alpha\right|}{\alpha} \times 100$$

where $R^{C^1_R}$ and $R^{C^2_R}$ are the feature subsets selected from the top or bottom α portion of the ranked feature rankings $R^{C^1_R}$ and $R^{C^2_R}$. The number is multiplied by 100 to get a percentage value.

4.5. Ensemble Feature Ranking

Ensemble learning methods have been widely used to address multiple issues in data mining and machine learning such as improving accuracy, generalization, and robustness of the learning model. Ensemble classification methods are learning algorithms that construct a set of classifiers and classify new data points by taking votes of their predictions. Dietterich (2000) have demonstrated that an ensemble of classifiers that have independent errors improves the overall accuracy. They have identified three main reasons to explain the positive effect of ensemble methods on classification: reducing the risk of choosing the wrong classifier, lowering the chance of getting stuck in local optima, and expanding the space of representable functions. Intuitively, the above reasoning should also hold in feature ranking with SVCs.

Based on the nature of the classifiers that participate in the voting, there could be two types of classifiers ensembles:

- Homogeneous ensemble of classifiers where all of the classifiers involved are of the same type and the training samples of each classifier are different. e.g., decision tree bagging.

- Heterogeneous ensemble of classifiers where the classifiers in the ensemble are not the same, for example when the final prediction result is produced from a combination of SVM, KNN, and DT prediction. This method is also referred to as consensus learning.

Since we want to study the effect of the ensemble approach on the SVC feature ranking bias, we have used a heterogeneous ensemble method. Formally, if $R^{C_R}(f'_i)$ which is the ranking of f'_i using SVC with classifier C_R is calculated based on the classification performance of the feature f'_i with the classifier C_R as $\lambda_{C_R}(X', \{f'_i\}, Y)$ then the ensemble ranking score of the feature f'_i (λ_{ens}) can be calculated based on the following:

$$\lambda_{ens}(X', \{f'_i\}, Y) = \tau\left(\lambda_{C^1_R}(X', \{f'_i\}, Y), \ldots, \lambda_{C^n_R}(X', \{f'_i\}, Y)\right)$$

where C^i_R indicates the classifiers used in the study and τ is the ensemble function.

There are multiple ensemble functions to combine the feature ranking results. Maximum, sum, mean, median, and different voting schemes which have been used and studied extensively (Chrysostomou et al., 2008, Makrehchi and Kamel, 2007, Yan, 2007). However, our study is not focused on comparing ensemble functions. Since the mean and the median are more commonly used ensemble functions, and an outlier can seriously distort the result of the mean (Kittler et al., 1998), the median has been adopted in our approach as the ensemble function. Furthermore, preliminary results of our experiments did not show significant outcome changes when substituting the median with the mean.

4.6. Bias Evaluation

To address the third question, we have selected feature subsets from the top of the feature ranking list and built classifiers with them. The performance of these classifiers have been evaluated via cross validation. Intuitively, the feature subsets selected from the SVC rankings generated by the same classifier (C_R) as the final learning classifier (C_L) (i.e., $C_R = C_L$) should perform better than the SVCs rankings generated with the other classifiers (i.e., $C_R \neq C_L$).

We should also consider that the threshold for choosing the feature subset changes the final results. To mitigate this effect, we have considered two scenarios. In the first scenario, we have chosen feature subsets containing different numbers of features. Feature subsets with the same number of features are then sorted based on the performance of the classifiers built with them and received placements from 1 indicating the best performance to n, the worst performance. This method is similar to comparing classifier performance which has been explained by Demsar (2006).

In other words, we have made subsets having 1 feature to subsets having α features and in every subset size, we have recorded the subset that comparatively works better with a specific classifier. Then, we have averaged the placements over the subsets with different sizes for all of the datasets. Since mean is sensitive to outliers, they are removed prior to calculating the average. To remove the outliers only 90% of the placements have been used and the least 5% and the greatest 5% of the values have been discarded. The pseudo-code that calculates the average placement values is shown in Algorithm 1. In this algorithm number of SVC ranking equals the number of classifiers (C_R) plus one for the ensemble ranking.

Algorithm 1 Average position of C_R ranking in C_L.

```
int c = Number of classifiers in the study;
int M.AUC[c,c+1,α], M.POSITION[c,c+1,α], M.MEAN.POSITION[c,c+1];
for i:=1 to α do
    for every "classifier" C_L do
        for every "SVC Ranking list" R^{C_R} do
            Feature.Subset=R_i^{C_R};
            Build C_L using Feature.Subset;
            M.AUC[C_L,C_R,i]=C_L × AUC;
        end for
        M.POSITION[C_L,,i]=order(M.AUC[C_L,,i]);
    end for
end for
for every "classifier" C_L do
    for every "SVC Ranking list" R^{C_R} do
        M.MEAN.POSITION[C_L,C_R]=
        mean(M.POSITION[C_L,C_R,],"Remove 10% Outliers");
    end for
end for
```

Although in practice, the feature subset selection threshold is mostly chosen arbitrary, it is possible to choose the best performing number of features in a subset, using a learning curve, and build a classifier with that number of features (Arauzo-Azofra et al., 2011). For this reason, we have also repeated the experiments by considering the number of features that result in the highest performance of all and repeated the above experiments without averaging over multiple thresholds.

5. Experimental Results

Seven classifiers and ten datasets have been used to conduct this study. Single variable classifiers have been built with each classifier on each dataset. Any samples with missing values have been eliminated prior to conducting the study. The classifiers used in the experiments are Support Vector Machines (SVM) with polynomial kernel, Naïve Bayes (NB), Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN) (K=5), Logistic Regression (LR), AdaBoost (AB) with decision stump meta classifier, and Random Forests (RF). The dataset used in the experiments are listed in Table 1 and have been obtained from University of California at Irvine machine learning data repository (UCI) (Frank and Asuncion, 2010), Arizona State University feature selection repository (ASU) (Liu and al., 2011), and Causality Workbench Repository (CWR) (CW-Team, 2011).

Table 1: Datasets used in the experiments.

Name	Samples	Features	Domain	Source
Internet Ads	3279	1558	Internet	UCI
BASE-HOOK	4862	1993	news-group	ASU
CINA	16033	133	Census	CWR
GLI 85	22283	85	Micro-array	ASU
MADE-LON	4400	500	Synthetic	UCI
MARTI	500	1025	Micro-array	CWR
MUSK (v.1)	476	166	Physics	UCI
REGED	500	1000	Micro-array	CWR
SMK	187	19993	Micro-array	ASU
SPECTF	267	44	Clinical	UCI

The experiments explained in the previous sections have been implemented using *Waikato Environment for Knowledge Analysis (WEKA)*, *Java* and *R*. For each feature in each dataset, a SVC has been built and evaluated via cross validation. The features have been ranked based on the AUC of the SVCs.

For example, Figure 3 demonstrates the ranking of 166 features from MUSK dataset that has been generated with SVC built with Naïve Bayes and KNN. Figure 3.(a) is the AUC of features SVCs and Figure 3.(b) is the rankings based on the SVC AUCs. Lines demonstrate the threshold for 30 top and bottom ranked features, and features marked with a triangle pointing up are top ranked features and the one marked with trian-

gles pointing down demonstrate features in the bottom section. For example, the top 30 ranking features with SVC built with Naïve Bayes are shown in boxes marked with 3, 6, and 9, and the bottom 30 are shown in boxes marked with 1, 4, and 7.

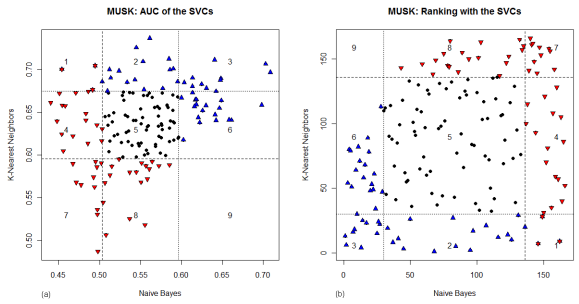

Figure 3: SVC feature ranking on MUSK using KNN and naïve Bayes: (a) AUC of SVC for features and (b) rankings based on the AUCs. Lines demonstrate the threshold for 30 top and bottom ranked features. Features marked with a triangle pointing up are top ranked features and the one marked with triangles pointing down demonstrate features in the bottom section.

The plot demonstrates how different the rankings with one classifier could be with respect to the other. Top 30 of the features ranked with KNN-SVC are in 1, 2, and 3 where NB-SVC's top 30 are in 3, 6, and 9. The only intersection between the two is box number 3 containing 11 features out of 30 which is only 37% agreement between the two. It is also interesting that they even disagree on the most discriminative feature. Boxes 1 and 9 contain features that are in one ranking method's selection list and in the other one's elimination nominees. As shown in the plot, there are 3 features in box 1 that NB sees as totally worthless while KNN ranks them as highly valuable.

It is trivial that the order of the features has not changed in the two scatter plots, although their distribution has. The distribution in Figure 3.(b) does not depend on the values of the AUCs which differ from one classifier to the other. AUC plot are more skewed towards the top which has no particular meaning in our context. It is only the variability of the rankings that is of interest in this study. Therefore, the rankings which are in more normalized form were used for correlation calculations.

This ranking was performed using all seven classifiers in SVCs. Figure 4 demonstrates the feature rankings from MUSK dataset generated with SVCs built using all seven classifiers in the study. It is seen that the rankings disagreement shown in Figure 3 for SVCs with NB and KNN, are present with most of the other classifiers as well. However, there are also some classifiers that generate more correlated ranking results.

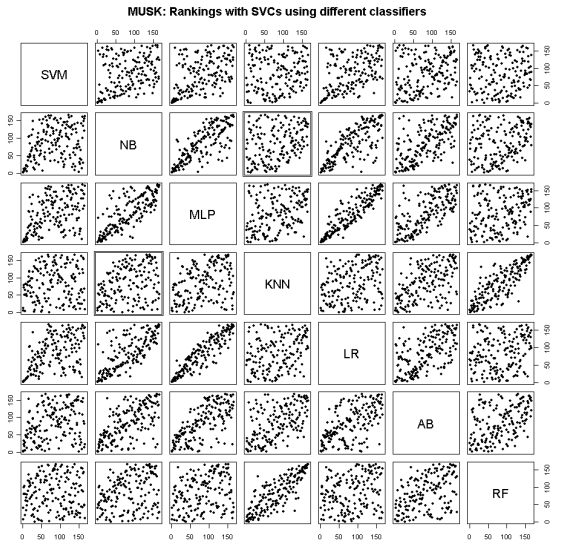

Figure 4: Feature rankings from MUSK dataset generated with SVCs built using all seven classifiers in the study. Plot from Figure 3 is highlighted using double borders.

Figures 3&4 demonstrate that SVC feature rankings generated using different classifiers are extremely different on the MUSK dataset. To quantify the results, we have calculated the Spearman's rank correlation measure for each two classifiers on all datasets. Table 2 summarizes the average correlation between the SVC rankings built with multiple classifiers over the ten datasets. It is seen that on average, the similarity between the results is only 0.60 and it is highly dependent on the dataset, considering the overall standard deviation of 0.24. However, it is interesting to note that the results from LR and MLP are highly correlated and NB demonstrates relatively high correlation with these two. KNN and RF on the other side produce relatively similar results and AB tends more towards the LR group. SVC rankings using SVM does not show strong correlation with any other classifier.

To see how the feature subsets selected using these SVCs are similar to each other, we have selected feature subsets from the top and bottom of the ranking lists. A fixed number of 30 features which is close to the average of 10% of the number of features in each dataset has been chosen to form the feature subsets. We calculated the similarity measure for the feature subsets selected using the 30 top and bottom features of the lists using the intersection percentage described earlier.

Table 3 shows similarities of the feature subsets. It

Table 2: Spearman correlation of the SVCs using different classifiers.

NB	0.54± 0.22	↩ρ				
MLP	0.63± 0.25	0.83± 0.10	↩ρ			
KNN	0.35± 0.28	0.53± 0.35	0.51± 0.34	↩ρ		
LR	0.67± 0.27	0.83± 0.13	0.94± 0.03	0.52± 0.36	↩ρ	
AB	0.48± 0.20	0.74± 0.20	0.73± 0.19	0.66± 0.27	0.71± 0.20	↩ρ
RF	0.29± 0.27	0.44± 0.35	0.44± 0.35	0.83± 0.12	0.43± 0.37	0.57± 0.28
$\mu\pm$ σ	SVM	NB	MLP	KNN	LR	AB

Average Mean = 0.60, Average STD = 0.24

Table 3: Similarity (Intersection) of the feature subsets selected using the top and bottom features of the SVC rankings.

(a) Similarity of the top 30 features using different classifiers.

NB	62.4± 24.9	↩ρ				
MLP	59.0± 33.7	56.2± 26.5	↩ρ			
KNN	64.1± 24.0	74.5± 23.6	56.9± 28.7	↩ρ		
LR	46.6± 32.0	56.2± 36.5	54.5± 31.2	56.5± 35.4	↩ρ	
AB	75.2± 29.8	77.6± 22.5	57.2± 31.1	82.4± 14.4	57.2± 37.3	↩ρ
RF	52.8± 27.8	63.8± 32.3	63.4± 28.8	63.4± 34.1	63.8± 33.9	65.2± 33.0
$\mu(\%)\pm$ $\sigma(\%)$	SVM	NB	MLP	KNN	LR	AB

Average Mean=61.5%, Average STD=30.7%

(b) Similarity of the bottom 30 features using different classifiers.

NB	25.7± 25.3	↩ρ				
MLP	35.3± 39.5	19.7± 25.5	↩ρ			
KNN	27.0± 29.4	47.3± 28.1	18.7± 27.5	↩ρ		
LR	19.3± 25.7	34.7± 39.1	23.7± 28.9	32.0± 36.9	↩ρ	
AB	41.3± 37.7	58.0± 23.9	24.0± 27.8	61.7± 25.5	37.3± 41.2	↩ρ
RF	20.7± 26.1	40.3± 35.4	21.3± 29.3	38.0± 36.6	39.0± 36.5	41.3± 36.7
$\mu(\%)\pm$ $\sigma(\%)$	SVM	NB	MLP	KNN	LR	AB

Average Mean=36.2%, Average STD=31.3%

is interesting to note that selected subsets from the top of the list are more similar than the ones from the bottom of the list. This might suggest that SVCs are more consistent in finding good quality features compared to distinguishing useless features. Therefore, classifiers might work better when used in forward feature selection than backward elimination. On the other hand, this phenomenon might be the result of abundance of useless features in the datasets which have no superiority against each other.

The results have a high standard deviation suggesting the influence of datasets on the outcome. It is also interesting that the two clusters of classifiers based on overall ranking correlation reported in the previous section do not exist here. Instead KNN, AB, and NB seem to be more similar in the top section. SVM and AB also show similar results. In the feature subsets from the bottom of the ranking list, the same similarity exists with less strength.

From the above experimental results, we can answer the first two questions. The overall average correlation of 0.6 and the overall average similarity of 61.5% in the top, and 36.2% in the bottom ranking feature subsets, suggests that rankings would be highly different when different classifiers are used to build the SVCs. As shown in the MUSK dataset, in many cases the results do not even agree on finding the most discriminative feature. This suggests that even wrappers that only select one top ranking feature for the next step of their algorithms might result in highly different selected feature subsets.

With regards to the second question, although some classifiers form groups looking at their correlations, these groups do not exist with respect to the top and the bottom ranking feature subsets. Therefore, it could be inferred that although there are some correlations between the overall rankings which is reported in Table 2, classifiers select the top ranking features differently. However, it should be noted that changing the thresholds for selecting the feature subsets changes the similarity results to some extent.

To address the third question and observe if there is any classifier bias in the SVC feature rankings, we have chosen feature subsets containing from 1 to 30 features using SVC rankings from all classifiers (C_R). Then, these subsets have been used to build the final classi-

fiers (C_L) and their performances evaluated via 5-fold cross validation in terms of AUC. For example Figure 5 demonstrates the performance of top 30 features from MUSK dataset based on different SVC rankings evaluated by NB, MLP and LR classifiers (C_L). Each point indicates the number of top performing features from each SVC rankings that the final classifier (C_L) was built with. It could be seen that in this dataset NB works best with its own SVC ranking on most points, KNN and RF share the best performing points when MLP is the final classifier (C_L), and SVM ranking works very well with LR in this dataset.

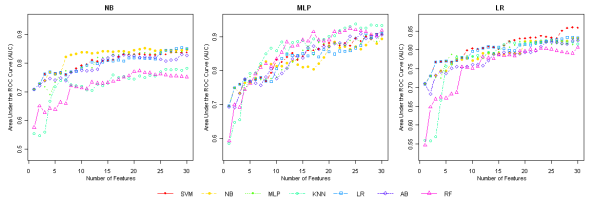

Figure 5: NB, MLP and LR classification performance of top 30 ranked features from MUSK dataset. Different feature rankings were obtained using all classifiers in the studies.

To generate more general results, the SVC rankings have been sorted based on the AUCs and placed from 1 for the best performance to 8 for the worst performance. The overall average placement numbers for each classifier on all datasets are shown in Figure 6.(a). For the second scenario where only the best performing number of features in a subset between 1 and 30 are chosen, the findings are shown in Figure 6.(b).

It could be seen that in Figure 6.(a) four out of seven classifiers do not generate the best classification performance with the SVC feature rankings using the same classifier. NB, LR and AB are the classifiers that show best performance when ranking and classification are done with the same classifiers. However, when considering the best number of features, the SVC ranking with LR becomes the second choice to be used with the LR classifier. SVM, MLP, KNN, and RF are classifiers that do not generate the best performance when used with an SVC ranking of the same type.

This experiment shows that some classifiers highly affect the SVC feature ranking, while some others do not. Therefore, depending on the classifier used, the SVC ranking will not be generated based on pure discriminative value of the features, and other measures should be considered. On the other hand, when the classifier that is to be used in building the final model (C_L) is known, such bias is desirable, since using a ranking that results in best performance in a classifier is the goal.

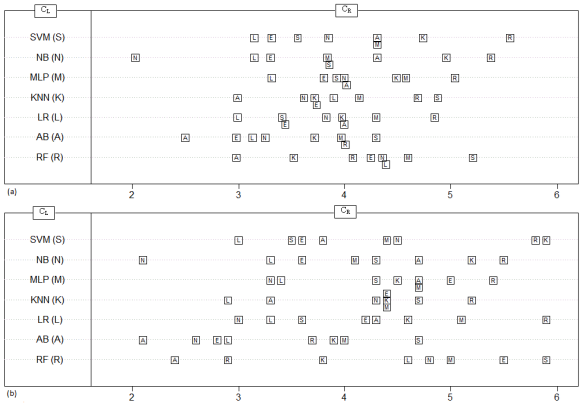

Figure 6: Average classification performance placement of different classifiers used in the SVC ranking (C_R) on all datasets. Each row corresponds to one classifier being the final learning classifier (C_L) and comparative performance placements are shown on the right side of the row. (a) Several numbers of features are selected for feature subsets and the results are averaged. (b) Only the best numbers of feature that generated the highest classification performance are considered in the study.

The above experiment, however, suggests that it is not always the best choice to use the same classifier in the SVC ranking (C_R) and final classification (C_L).

Even in cases where using the same classifiers in both stages result in comparative best performance, the average placement is not near the first. This suggests that even though on average they are better than other classifiers used in SVCs, they are not always the best choices and other choices generate better results depending on the dataset. For the other four cases, even on average, it does not produce the best performance to use the same classifier in the SVC ranking and final classification.

To address the fourth question about the ensemble ranking, we have included the ensemble SVC ranking in Figure 6 charts, shown with an E symbol. It could be seen that in Figure 6.(a) which shows that average number of features selected, the ensemble SVC ranking is mostly placed as the second or third SVC ranker. It performs relatively worst in Figure 6.(b) when the best numbers of features are considered. In these experiments, RF seems to be an exception where ensemble SVC ranking performs near the worst. Although ensemble rankings never generate the best performance with any classifier, on average it performs reasonably well when the best numbers of features to choose for a subset are unknown, thus making it a good candidate for ranking. It should also be noted that in these experiments, only the first 30 top performing features are considered, while the true best number of feature might be beyond 30.

Knowing that the de facto standard of using the same classifier for SVC and final classifier will not always generate the best results, we would like to understand how far from the optimal choice of SVC ranker is the final classification performance for each classifier if such approach is taken. The fifth question that we try to answer in the paper addresses this concern.

In these experiments, for each number of features between 1 and 30, we have calculated the distance between the best performances with that number of features (best combination of C_R and C_L), and the performance of using the same classifier in the SVC ranking (i.e., $C_R = C_L$). By removing the 10% outliers similar to algorithm 1 the averages performance losses are reported in Table 4.(a) in terms of the AUC percentage. The smallest loss was with NB and AB which also in Figure 6 demonstrated the best performance while using the same classifiers in SVC (C_R) and final classification (C_L).

The largest loss is shown in KNN and RF which also in Figure 6 did not show good performances. This study suggests that by using different classifiers for SVC ranking and final classification a gain of 0.24% up to 5.56% in AUC might be achieved depending on the final classifier. The high variation of the results demonstrates the high dependency of this conclusion on the datasets. Table 4.(b) summarizes the results from the same type of experiment when the best numbers of features were chosen for the subsets. Similar but more extreme distances are reported in this table.

Table 4: Average performance loss from the optimal combinations when using the same classifier in the SVC feature ranking (C_R) and final classification task (C_L). The numbers show AUC percentages.

(a) Several numbers of features are selected for feature subsets and the results are averaged.

	SVM	NB	MLP	KNN	LR	AB	RF
AD	11.85	1.00	0.64	0.07	0.26	0.15	0.18
BASEHOOK	0.68	0.01	0.38	0.51	0.34	0.10	0.34
CINA	0.71	0.04	0.09	0.09	0.02	0.08	0.14
GLI	2.97	0.22	1.61	2.07	4.14	1.01	1.75
MADELON	0.13	0.25	2.39	27.59	0.34	1.64	16.84
MARTI	2.39	0.00	6.45	17.41	10.60	2.58	18.58
MUSK	0.09	0.21	4.59	0.29	1.43	3.15	1.21
REGED	0.34	0.39	0.15	0.17	1.10	0.02	0.28
SMK	0.11	0.32	6.87	3.66	0.91	0.29	8.20
SPECTF	1.52	0.00	3.03	3.73	4.98	1.18	1.69
Average Distance(%)	2.08± 3.39	0.24± 0.29	2.62± 2.43	5.56± 8.87	2.41± 3.17	1.02± 1.07	4.92± 6.79

(b) Only the best numbers of features that generated the highest classification performance are considered in the study.

	SVM	NB	MLP	KNN	LR	AB	RF
AD	10.30	2.02	1.03	0.00	0.41	0.00	0.03
BASEHOOK	0.54	0.10	0.10	0.21	0.19	1.01	0.89
CINA	1.54	0.70	0.23	0.00	0.40	0.53	0.00
GLI	1.12	0.00	5.78	6.78	2.13	2.91	0.00
MADELON	0.00	0.29	3.49	28.22	0.00	0.27	18.22
MARTI	0.00	0.00	12.05	24.10	25.90	0.00	20.86
MUSK	0.00	0.92	4.26	0.96	3.14	3.39	0.00
REGED	0.38	0.43	0.00	0.00	0.88	0.02	0.05
SMK	0.00	0.00	13.41	4.94	3.14	1.55	7.45
SPECTF	4.97	0.00	5.61	9.35	3.72	0.00	0.00
Average Distance(%)	1.88± 3.15	0.45± 0.61	4.60± 4.58	7.46± 9.90	3.99± 7.72	0.97± 1.20	4.75± 7.73

Table 5: AUC of the different classifiers (C_L) when trained and tested with the first eight and night features from the KNN (C_R) feature ranking. *F121* improves the performance of KNN classifiers (C_L) while exacerbating the NB performance.

Rank	Feature	SVM	NB	MLP	KNN	LR	AB	RF
8	F98	0.774	0.723	0.849	0.858	0.779	0.796	0.909
9	F121	0.772	0.720	0.866	0.868	0.777	0.796	0.924

6. Discussion

To understand the reason behind the phenomena in Figure 3 where NB places a feature in its bottom list and KNN ranks the same feature in the top performing category, we have studied such features from the MUSK dataset in more details. Feature *F121* which according to the description in the UCI repository (Frank and Asuncion, 2010) is a kind of *distance measurement*, is one of the three features with such characteristics in Figure 3. It is ranked as the ninth best performing feature using KNN while NB ranks it as one of the worst performing features. The effect of this feature in the final classification task is shown in Table 5. When KNN is used as the final classifier (C_L), *F121* increases the AUC by 1% and when NB is used as the final classifier *F121* reduces the AUC by 0.3%. By looking at the changes that *F121* brings to the performance of other classifiers it seems that the patterns in this feature are more local than global.

Histograms of instance values distributions based on *F121* feature are shown in the Figure 7.(a) and Figure 7.(b) for each class. It is also shown how they could be estimated via a kernel density estimator with a higher smoothing level that is closer to a Gaussian, and with a less smoothing hyper-parameter . In Figure 7.(c) it can be seen that there are not much difference between the near Gaussian distribution estimation of two classes while the less smooth kernel density estimation provides some distinctions around values of 0 and 100. This variation demonstrates the reason that KNN which captures the local features performs better than NB. In our scenario NB uses a fixed smoothing hyper-parameter for the distributions. By changing the density estimation method in NB, *F121* no longer performs among the worst features and provides average performance.

While the smoother distribution estimation works

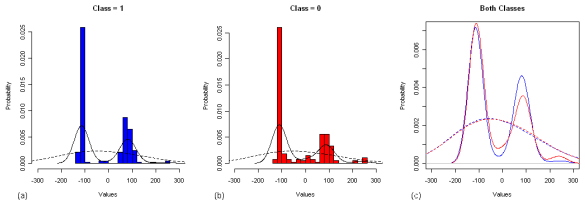

Figure 7: Histogram of values from *F121* for each class is shown in (a) and (b). Comparison of kernel density estimation with more smooth and less smooth hyper-parameter is shown in (c) for both classes where blue indicates class=1 and red indicates class=0.

well with other features, this performance change implies that the fixed hyper-parameters chosen for NB is not suitable for the *F121* feature. However, it is virtually not possible to tune the hyper-parameters for each feature using a classifier as a means to measure the quality of features. Since a separate classifier is built for every feature, the model selection problem should be addressed to build the perfect classifier for that feature. Depending on the classifier (C_R), running a grid search or other methods to find the best possible hyper-parameters for each feature is simply not possible due to extreme time consumption. The common way is to set the hyper-parameters to values that work best on average or use the hyper-parameters considered for the final classifier (C_L). This introduces the challenge of how to find the best hyper-parameter of the final classifier before knowing the final features. Hence, we argue that when using a classifier to find the best features, the search is not only in the feature space but also in the hyper-parameter space at each step. Wrappers and Single Variable Classifiers both suffer from this issue.

7. Conclusion and Future Directions

In this paper we investigated the set of questions we highlighted in section 4 regarding the bias and stability of single variable classifier feature ranking using multiple classifiers and datasets. The most important findings include:

1. We showed that SVC feature ranking is highly sensitive to the choice of classifiers. Average correlation of overall feature rankings generated using different classifiers is only 0.60, more importantly the SVC rankings only agree on 61.5% of the 30 most predictive features, and on 36.2% of the 30 least predictive features. This finding strongly suggests that SVC feature rankings are not good candidates to report predictive power of features, e.g. in medical and biological domains. It also suggests that SVC rankings are even less robust in backward feature elimination methods comparing to forward feature selection approaches due to their high disagreement rate for elimination (i.e. only 36.2% agreement)
2. While the effect of classifiers on SVC ranking is expected, we also empirically challenged the de facto standard of using the same classifier for ranking and final classification ($C_R = C_L$) and its efficiency to generate the best prediction performance. In four out of seven classifiers used in our study, the best classification performances were not achieved when using the same classifiers in the rankings and final classification tasks ($C_R = C_L$).
3. Furthermore, we have shown the comparative loss of performance when following the de facto standard of using the same classifier in the SVC ranking and final classification with respect to the optimal classifier combination for ranking and classification. We have observed that NB and AB classifiers show insignificant loss of performance comparing to other combinations and are good candidates to be used for both roles ($C_R = C_L$). In contrast, KNN and RF suffered the highest loss of performance when used for both ranking and classification comparing to better combinations. While providing insights about the stability of each classifier with respect to feature rankings, this study suggests that NB and AB may be better candidates for SVC rankings and classification, comparing to KNN and RF classifiers.
4. Several methods referenced in section 3 choose an ensemble approach towards feature selection and ranking both by combining different classifiers or using same classifiers with different settings. We have demonstrated that ensemble feature ranking by using different classifiers would not generate the overall best feature rankings but it is a ranking method that generates above average performance with most classifiers. This ensemble approach can provide a more independent insight when investigating the predictive value of each feature, e.g. in medical and biological domains.

Although we have performed multiple studies investigating single variable classifier feature rankings, there are several other settings that could affect the performance of SVCs. It would be highly valuable to study how data distributions in each feature affect the performance of the overall SVC feature rankings. Furthermore, the effect of different ensemble settings that can

be generated using different sampling of data is interesting to study and can provide better insights to ensemble feature rankings. Our studies show that there is no best performing feature subset for different classifiers and suggest using different feature rankings and selections for each weak learner (classifier) in heterogeneous ensemble methods. An ensemble method that could use this diversity may achieve high performance and is worth studying.

In section 6 we provided an example to show the significant effect of not only the classifier type but also its hyper-parameters on the performance of SVC feature ranking and the overall performance. This issue affects both SVC feature rankings and more widely used wrapper methods. Our studies suggest that when using a classifier to find the best features, both feature space and hyper-parameter space should be searched at each step. Finding the best feature subset depends on a specific hyper-parameter setting for a classifier, while the common approach of finding the best values for hyper-parameters are via cross-validation using a certain set of features. Therefore, separating the feature ranking/selection and hyper-parameter tuning steps fails to capture the best combinations between them. Zhao et al. (2011) and Yoon et al. (2013) show enhanced performance by combining feature selection and parameter optimization for SVMs and Neural Networks. More systematic and general approaches could be highly valuable in finding better combinations and achieving higher performances.

Acknowledgments

This work was supported in part by NIH grants R01-EB002450 and R01-EB013227.

References

Isabelle Guyon. Practical feature selection: from correlation to causality. *Mining massive data sets for security: advances in data mining, search, social networks and text mining, and their applications to security*, pages 27–43, 2008.

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

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–82, 2003.

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

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2nd edition, February 2009.

Zheng Alan Zhao and Huan Liu. *Spectral feature selection for data mining*. Chapman & Hall/CRC, 2011.

Steven Loscalzo, Lei Yu, and Chris Ding. Consensus group stable feature selection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 567–575. ACM, 2009.

J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Verlag, 2007.

K. Kira and L. A. Rendell. The feature selection problem: traditional methods and a new algorithm. In *AAAI-92. Proceedings Tenth National Conference on Artificial Intelligence*, pages 129–34. AAAI Press, 1992.

Roberto Ruiz, Jose C. Riquelme, and Jesus S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39:2383–2392, 2006.

Herve Stoppiglia, Gerard Dreyfus, Remi Dubois, and Yacine Oussar. Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, 3:1399–1414, 2003.

I Slavkov, B Zenko, and S Dzeroski. Evaluation method for feature rankings and their aggregations for biomarker discovery. *Journal of Machine Learning Research*, 2010.

J. Demsar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Antonio Arauzo-Azofra, Jos Luis Aznarte, and Jos M. Bentez. Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, 38:8170–8177, 2011.

I. Guyon, H.M. Bitter, Z. Ahmed, M. Brown, and J. Heller. Multivariate non-linear feature selection with kernel methods. *Soft Computing for Information Processing and Analysis*, pages 313–326, 2005.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

Yonghong Peng, Zhiqing Wu, and Jianmin Jiang. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43:15–23, 2010.

M. Bacauskiene, A. Verikas, A. Gelzinis, and D. Valincius. A feature selection technique for generation of classification committees and its application to categorization of laryngeal images. *Pattern Recognition*, 42:645–54, 2009.

Iffat A. Gheyas and Leslie S. Smith. Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43:5–13, 2010.

A. Tsanas, M.A. Little, and P.E. McSharry. A simple filter benchmark for feature selection. *Journal of Machine Learning Research*, 2010.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:301–320, 2005.

K. Chrysostomou, S. Y. Chen, and Liu Xiaohui. Combining multiple classifiers for wrapper feature selection. *International Journal of Data Mining, Modelling and Management*, 1:91–102, 2008.

Shobeir Fakhraei, Hamid Soltanian-Zadeh, Farshad Fotouhi, and Kost Elisevich. Consensus feature ranking in datasets with missing values. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, pages 771–775. IEEE, 2010a.

Gokhan Gulgezen, Zehra Cataltepe, and Lei Yu. Stable and accurate feature selection. In *Lecture Notes in Computer Science*, volume 5781 LNAI, pages 455–468. Springer Verlag, 2009.

A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12:95–116, 2007.

Lei Yu, Chris Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 803–811. ACM, 2008.

Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Lecture Notes in Computer Science*, volume 5212, pages 313–325. Springer Berlin / Heidelberg, 2008.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

Jana Novovicová, Petr Somol, and Pavel Pudil. A new measure of feature selection algorithms' stability. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 382–387. IEEE, 2009.

Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

L. Li, J. Zhang, and R.M. Neal. A method for avoiding bias from feature selection with application to naive bayes classification models. *Bayesian Analysis*, 3:171–196, 2008.

Surendra K. Singhi and Huan Liu. Feature subset selection bias for classification learning. In *ACM International Conference Proceeding Series*, volume 148, pages 849–856. ACM, 2006.

Mark A. Hall and Geoffrey Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15:1437–1447, 2003.

W. Altidor, T. M. Khoshgoftaar, and J. Van Hulse. An empirical study on wrapper-based feature ranking. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2009)*, pages 75–82. IEEE, 2009.

Masoud Makrehchi and Mohamed S. Kamel. Combining feature ranking for text classification. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pages 510–515. IEEE, 2007.

Weizhong Yan. Fusion in multi-criterion feature ranking. In *Information Fusion, 2007 10th International Conference on*, pages 1–6. IEEE, 2007.

Fei-Long Chen and Feng-Chia Li. Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37:4902–4909, 2010.

R. Bryll, R. Gutierrez-Osuna, and F. Quek. Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36:1291–302, 2003.

Shobeir Fakhraei, Hamid Soltanian-Zadeh, Farshad Fotouhi, and Kost Elisevich. Effect of classifiers in consensus feature ranking for biomedical datasets. In *Proceedings of the ACM fourth international workshop on Data and text mining in biomedical informatics*, pages 67–68. ACM, 2010b.

Laura Emmanuella A dos S Santana and Anne M Canuto. Filter-based optimization techniques for selection of feature subsets in ensemble systems. *Expert Systems with Applications*, 41(4, Part 2):1622 – 1631, 2014.

Rafael M.O. Cruz, George D.C. Cavalcanti, Ing Ren Tsang, and Robert Sabourin. Feature representation selection based on classifier projection space and oracle analysis. *Expert Systems with Applications*, 40(9):3813 – 3827, 2013.

Combination of single feature classifiers for fast feature selection. In *Advances in Knowledge Discovery and Management*, volume 527, pages 113–131. Springer International Publishing, 2014.

Wenjie You, Zijiang Yang, and Guoli Ji. Feature selection for high-dimensional multi-category data using pls-based local recursive feature elimination. *Expert Systems with Applications*, 41(4, Part 1):1463 – 1475, 2014.

Feature extraction using single variable classifiers for binary text classification. In *Recent Trends in Applied Artificial Intelligence*, volume 7906, pages 332–340. Springer Berlin Heidelberg, 2013.

Hyunsoo Yoon, Cheong-Sool Park, Jun Seok Kim, and Jun-Geol Baek. Algorithm learning based neural network integrating feature selection and classification. *Expert Systems with Applications*, 40 (1):231 – 241, 2013.

Mingyuan Zhao, Chong Fu, Luping Ji, Ke Tang, and Mingtian Zhou. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, 38(5):5197 – 5204, 2011.

G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3: 1289–305, 2003.

T. Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31:138, 2004.

Xue-Wen Chen and Michael Wasikowski. FAST: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 124–132. ACM, 2008.

Rui Wang and Ke Tang. Feature selection for maximizing the area under the roc curve. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 400–405. IEEE, 2009.

T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems. First International Workshop, MCS 2000. Proceedings (Lecture Notes in Computer Science Vol.1857)*, pages 1–15. Springer-Verlag, 2000.

J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20:226–239, 1998.

A. Frank and A. Asuncion. UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences* {*http://archive.ics.uci.edu/ml*}, 2010.

Huan Liu and et al. ASU feature selection data repository. *Arizona State University* {*http://featureselection.asu.edu/*}, 2011.

CW-Team. Causality workbench repository {http://www.causality.inf.ethz.ch/repository.php}. 2011.