Stable and Accurate Feature Selection

Gokhan Gulgezen*, Zehra Cataltepe**, and Lei Yu

Istanbul Technical University, Computer Engineering Department, Istanbul, Turkey
Binghamton University, Computer Science Department, Binghamton, NY, USA
gulgezen@itu.edu.tr, cataltepe@itu.edu.tr, lyu@cs.binghamton.edu

Abstract. In addition to accuracy, stability is also a measure of success for a feature selection algorithm. Stability could especially be a concern when the number of samples in a data set is small and the dimensionality is high. In this study, we introduce a stability measure, and perform both accuracy and stability measurements of MRMR (Minimum Redundancy Maximum Relevance) feature selection algorithm on different data sets. The two feature evaluation criteria used by MRMR, MID (Mutual Information Difference) and MIQ (Mutual Information Quotient), result in similar accuracies, but MID is more stable. We also introduce a new feature selection criterion, MID_α, where redundancy and relevance of selected features are controlled by parameter α.

Keywords: Feature Selection, Stable Feature Selection, Stability, MRMR (Minimum Redundancy Maximum Relevance).

1 Introduction and Previous Work

Many feature selection algorithms have been developed in the past with a focus on improving classification accuracy while reducing dimensionality. Traditionally, the relevance of a feature is the most important selection criterion because using highly relevant features improves classification accuracy [1]. A majority of feature selection algorithms concentrate on feature relevance [2]. In order to have a more compact feature subset with good generalization, the selected features need to be non-redundant. There are several studies on feature redundancy and how the trade-off between feature relevance and redundancy affects classification accuracy [3,4].

A relatively neglected issue is the *stability of feature selection* - the insensitivity of the result of a feature selection algorithm to variations in the training set. This issue is important in many applications with high-dimensional data, where feature selection is used as a knowledge discovery tool for identifying characteristic markers for the observed phenomena [5]. For example, in microarray data analysis, a feature selection algorithm may select largely different subsets of features (genes) under variations to the training data [6,7]. Such instability dampens the confidence of domain experts in investigating any of the various

* Supported by Tubitak master scholarship.
** Supported partially by Tubitak research project no 105E164.

W. Buntine et al. (Eds.): ECML PKDD 2009, Part I, LNAI 5781, pp. 455–468, 2009.

subsets of selected features for biomarker identification. It is worthy noting that stability of feature selection results should be investigated together with classification accuracy, because domain experts are not interested in a strategy that yields very stable feature sets, but leads to a bad predictive model (e.g., arbitrarily picking the same set of features under training data variation).

There exist very limited studies on the stability of feature selection algorithms. An early work in this direction was done by Kalousis et al. [6]. Their work compared the stability of a number of feature ranking and weighting algorithms under training data variation based on various stability measures on high-dimensional data, and demonstrated that different algorithms which performed similarly well for classification had a wide difference in terms of stability. More recently, two techniques were proposed to explicitly achieve stable feature selection without sacrificing classification accuracy: ensemble feature selection [8] and group-based feature selection [7].

The above studies have not addressed an important issue: how different tradeoff between relevance and redundancy affects the stability of feature selection algorithms. Our study attempts to address this issue by evaluating the stability of two different MRMR (Minimum Redundancy Maximum Relevance) feature evaluation criteria: MID (Mutual Information Difference) and MIQ (Mutual Information Quotient), which balance the two objectives, maximum relevance and minimum redundancy, in different ways. We theoretically and empirically show that MID produces more stable feature subsets. Furthermore, we introduce an extension of MID where relevance and redundancy of a feature may have different weights in feature evaluation. We show that for each data set, stability of MRMR can be controlled through the use of this weighting parameter.

The rest of the paper is organized as follows: In Section 2 we review the MRMR feature selection method, and the two criteria MID and MIQ which are used by MRMR for feature evaluation. Section 3 discusses the stability measure that we use. Section 4 discusses theoretically and practically why MID is more stable than MIQ and introduce the extension MID_α where the contribution of redundancy and relevance to feature score calculation is scaled by means of a parameter α. In Section 5 experimental results are given and Section 6 concludes the paper.

2 MRMR Feature Selection Algorithm

MRMR [9] is a filter based feature selection algorithm which tries to select the most relevant features with the target class labels and minimize the redundancy among those selected features simultaneously, the algorithm uses Mutual Information $I(X,Y)$ that measures the level of similarity between two discrete random variables X and Y:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log(\frac{p(x,y)}{p_1(x)p_2(y)}) \quad (1)$$

where $p(x, y)$ is the joint probability distribution function of X and Y, and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of X and Y respectively.

For notational simplicity, we represent each feature f_i using the vector of N observations for that feature: $f_i = \left[f_i^1, f_i^2, f_i^3, ..., f_i^N\right]$. f_i is an instance of the discrete random variable F_i. $I(F_i, F_j)$ will be used to represent the mutual information between features i and j, where $i, j = 1, 2, ..., d$ and d is the input dimensionality which equals the number of features in the dataset. In order to measure relevance, MRMR algorithm again uses mutual information between target class label $h = \left[h^1, h^2, h^3, ..., h^N\right]$ and the feature i which will be denoted as $I(H, F_i)$.

Let S denote the feature set that we want to select and $|S|$ its cardinality. In order to make sure that the selected feature subset is the best subset, two conditions should be met. First one is the minimum redundancy condition:

$$W = \frac{1}{|S|^2} \sum_{F_i, F_j \in S} I(F_i, F_j) \tag{2}$$

and the other one is the maximum relevancy condition:

$$V = \frac{1}{|S|} \sum_{F_i \in S} I(F_i, H) \tag{3}$$

According to [9], the two simplest combinations of these two conditions are:

$$max(V - W) \tag{4}$$

$$max(V/W) \tag{5}$$

Because of the fact that obtaining the best subset that satisfies one of the above equations requires $O(N^{|S|})$ search, MRMR uses the following algorithm to solve this optimization problem. First feature is selected according to Eq. (3). After that the feature i that satisfies the conditions below in Eqs. (6) and (7) is selected at each step and the selected features remain in the feature set S. m is the number of features in feature set (number of selected features) and $\Omega_S = \Omega - S$ is the feature subset of all features except those already selected.

$$\min_{F_i \in \Omega_S} \frac{1}{|S|} \sum_{F_j \in S} I(F_i, F_j) \tag{6}$$

$$\max_{F_i \in \Omega_S} I(F_i, H) \tag{7}$$

The combination of Eqs. (6) and (7) according to Eqs. (4) and (5) result in two selection criteria in Table 1:

As it can be seen, Eq. (6) is equivalent to the condition in Eq. (2) and Eq. (7) is an approximation of Eq. (3). The complexity of the algorithm above is given to be $O(|S| \cdot N)$ in [9].

Table 1. Two different schemes to search for the next feature in MRMR optimization conditions

ACRONYM	FULL NAME	FORMULA
MID	Mutual information difference	$\max_{F_i \in \Omega_S} \left[I(F_i, H) - \frac{1}{\lvert S \rvert} \sum_{F_j \in S} I(F_i, F_j) \right]$
MIQ	Mutual information quotient	$\max_{F_i \in \Omega_S} \left\{ I(F_i, H) / \left[\frac{1}{\lvert S \rvert} \sum_{F_j \in S} I(F_i, F_j) \right] \right\}$

3 Stability Evaluation

In order to measure the stability of a feature selection algorithm, a measure of similarity between two sets of feature selection results is needed. We will use a method similar to the one proposed by [7]. Let $R_1 = \{F_i\}_{i=1}^{|R_1|}$ and $R_2 = \{F_j\}_{j=1}^{|R_2|}$ denote two sets of feature selection results and each F_i and F_j represent an individual feature. In order to evaluate stability between R_1 and R_2, [7] propose to model R_1 and R_2 together as a weighted complete bipartite graph $G = (V, E)$, with nodes $V = R_1 \cup R_2$, and edges $E = \{(F_i, F_j) \mid F_i \in R_1, F_j \in R_2\}$, and every edge (F_i, F_j) is associated with a weight $\omega(F_i, F_j)$. In our method, all weights are determined by calculating the symetrical uncertainty between pair of features F_i and F_j. This entropy based nonlinear correlation is called symmetrical uncertainty, SU, and is calculated in the following way:

$$SU_{i,j} = 2 \left[\frac{IG(F_i \mid F_j)}{H(F_i) + H(F_j)} \right] \tag{8}$$

As it is defined earlier, in this equation, F_i and F_j are random variables which refer to the i th and jth input features respectively and information gain, entropy and conditional entropy are defined as:

$$IG\left(X \mid Y\right) = H\left(X\right) - H\left(X \mid Y\right) \tag{9}$$

$$H\left(X\right) = \sum_{x \in X} p\left(x\right) log_2\left(p\left(x\right)\right) \tag{10}$$

$$H\left(X \mid Y\right) = \sum_{y \in Y} p\left(y\right) \sum_{x \in X} p\left(x \mid y\right) log_2\left(p\left(x \mid y\right)\right) \tag{11}$$

In our method $\omega\left(F_i, F_j\right)$ equals to $SU_{i,j}$ and the overall similarity between feature sets R_1 and R_2 is defined as:

$$Sim^M(R_1, R_2) = \frac{1}{|M|} \sum_{F_i, F_j \in M} \omega(F_i, F_j) \tag{12}$$

where M is a maximum matching in G. The problem of maximum weighted bipartite matching (also known as the assignment problem) is to find an optimal matching where the sum of the weights of all edges in the matching has a maximal value. There exist various efficient algorithms for finding an optimal solution. The purpose of using such a method is to assess the similarity between two sets of features by considering the similarity of feature values instead of features indices which makes sense when two feature subsets contain a large portion of different but highly correlated features. In order to find the optimal solution, Hungarian Algorithm is used. The algorithm is implemented by Alexander Melin from University of Tennessee and taken from the Matlab Central web site [10]. The algorithm is designed for finding minimum weight matching so in order to find maximum weight matching, the sign of the entries of the performance matrix is inversed.

Our stability measure differs from that of Yu and Ding in the following way. First of all, in their paper, they measure the similarity between two sets of feature groups not two sets of individual features. Second, each weight in a bipartite graph is decided by the correlation coefficient between the centers or the most representative features of the two feature groups. Our methodology is a special case of that of Yu and Ding where each F_i and F_j represent individual features and the similarity between them is decided by the symmetrical uncertainty.

4 Stability of MID versus MIQ

In this section, the stability of MID and MIQ techniques are compared both theoretically and experimentally.

4.1 Theoretical Analysis

As it is mentioned before, the MRMR algorithm uses two basic calculations, MID (Mutual information difference) and MIQ (Mutual information quotient), for selecting the next feature among Ω_S, the feature subset of all features except those already selected. MID is the difference of the mutual information between feature F_i and the class label h and the mean of the sum of mutual information values between feature F_i and F_j, which $F_j \in S$ and $j = 1, 2, \ldots, |S|$. MIQ is the ratio of the mutual information between feature F_i and the class label h to the mean of the sum of mutual information values between feature F_i and F_j, which $F_j \in S$ and $j = 1, 2, \ldots, |S|$. Since we have a limited number of samples, we can not obtain the real probability distribution of features or labels, therefore the mutual information values computed also contain a sampling error. Since both features and labels are discrete random variables, the feature-feature or feature-label mutual information computations contain similar types of error.

Let $V+\epsilon$ and $W+\delta$ be the random variables that correspond to the relevance and redundancy values computed over a sample of size N and let ϵ and δ be distributed as $N(0, \sigma_N^2)$.

The variance of MID and MIQ is a direct indication of the stability of these estimators, because as the variance increases different features could be selected at each step of feature selection.

The variance of MID is quite easy to compute:

$$var(MID) = var((V+\epsilon) - (W+\delta)) = 2\sigma_N^2 \tag{13}$$

The mean of MID equals the actual value of the difference between relevance and redundancy.

The mean and variance of the ratio MIQ is much harder to compute. First of all, if $W+\delta$ has a nonnegligible distribution around 0, then the ratio has a Cauchy component, which means the mean and variance are undefined and the second moment is infinite [11]. When both the numerator and the denominator are far from zero, then the ratio is normally distributed with mean V/W and unit variance. As seen in the next section in the experimental analysis, especially for small number of features, W is close to zero and the MIQ shows a large variance.

4.2 Experimental Analysis

Since we have only a finite number of samples, computation of the actual values of V and W are not possible. In order to estimate the mean and variances of V, W, MID and MIQ, we use bootstrapping. We apply MID and MIQ

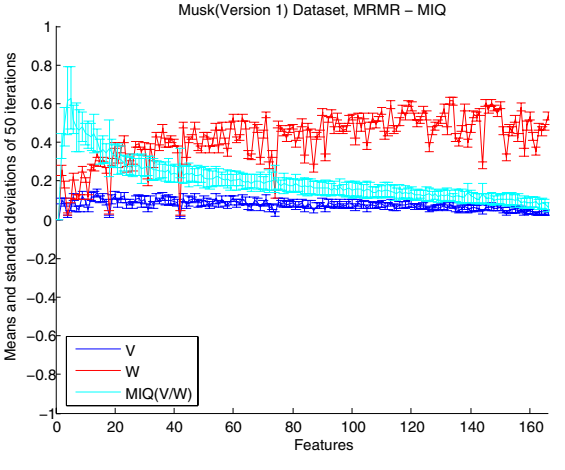


Fig. 1. Mean and standard deviation of the V, W and V/W (MIQ) calculations(y axis) while the number of selected features(x axis) varies for Musk (Version 1) dataset

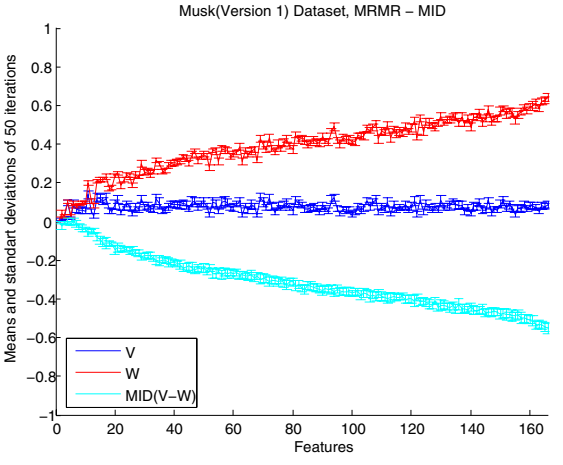


Fig. 2. Mean and standard deviation of the V, W and $V - W$ (MID) calculations(y axis) while the number of selected features(x axis) varies for Musk (Version 1) dataset

techniques respectively on the whole dataset and obtain feature sequences for each algorithm, then we bootstrap the data and do the mutual information calculations again on these new datasets according to feature sequences that are obtained before. We repeat the bootstrap evaluations 50 times and present mean and standart deviation of the values in each feature selection step for the Musk(Version 1) dataset from the UCI [12].

As seen in figures 1 and 2, the entropy values that are calculated by MID technique have smaller variance than the entropy values that are calculated by MIQ, which means difference of V and W gives more stable results.

4.3 MID_α Trading Off Stability and Accuracy

As seen at the previous section, MID gives more stable results than MIQ. We propose a modification to MID as follows: $MID_\alpha = \alpha V - (1 - \alpha) W$. We aim to control stability and accuracy by means of changing α. Various α values($\alpha = [0, 0.25, 0.5, 0.75, 1]$) are used in the Results section below.

5 Experiments

In this section we give details on the datasets used in the experiments, experimental setup and results.

5.1 Data Sets

Experiments were performed on 8 datasets. Seven of them were from the UCI [12]: Ionosphere, Sonar, Parkisons, Musk (Version 1), Multiple-features, Handwritten digits and Wine. The eighth dataset was the Audio Genre data set.

Audio genre dataset consists of the 5 least confused genres of the Tzanetakis data set [13]: Classical, Hiphop, Jazz, Pop and Reggae, each with 100 samples. Two different sets of audio features are computed. First Timbral, rhytmic content and pitch content features yielding 30 features are extracted using Marsyas toolbox [13]. Second, 20 features covering temporal and spectral properties are etracted using the databionic music miner framework given in [14].

All feature values in the datasets are discretized to 10 equal length bins between their maximum and minimum values. MRMR algorithm executions and stability calculations between feature sequences are performed on these discretized features.

Table 2 shows the number of features, instances and classes for the 8 datasets.

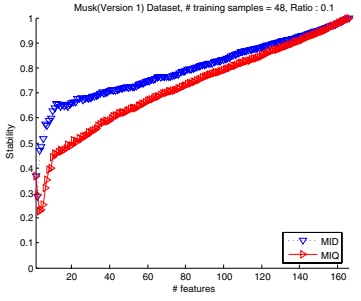
Table 2. Information about datasets

Dataset	Features	Instances	Classes
Ionosphere	34	351	2
Sonar	60	208	2
Parkinsons	23	195	2
Musk(Version 1)	166	476	2
Audio Genre	50	500	5
Multi-features Digits	649	2000	10
Handwritten Digits	64	3823	10
Wine	13	178	3

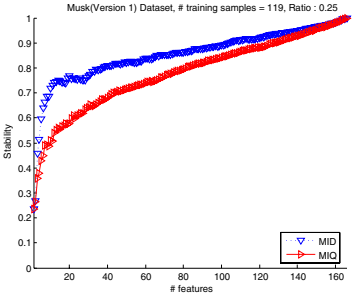
5.2 Experimental Setup

The stability of a feature selection algorithm is defined as the average similarity of various sets of results produced by the same feature selection algorithm under training data variations. Each subset of samples can be obtained by randomly sampling or bootstrapping the full set of samples. Lets say, we have a dataset D which contains N samples and we want to measure the stability of a specific algorithm. In order to do that, we bootstrap N samples from the dataset 10 times to obtain 10 training sets, $D_{train,i}$, $i = 1, 2, \ldots, q$ where $q = 10$. After each bootstrap process, samples that do not belong to $D_{train,i}$ are considered to be the test set $D_{test,i}$ for that iteration. In order to demonstrate the stability of an algorithm, first, a feature selection is performed using all training data and feature sequence R_i is obtained. Since we want to compare the change in the selected features when the sample size gets smaller, we draw samples of size $r * |D_{train,i}|$ from $D_{train,i}$ where we chose r from $r = \left[r^1, r^2, \ldots, r^j\right]$. In the experiments below we chose $j = 5$ and $r = [0.1, 0.25, 0.5, 0.75, 1]$. We obtain feature sequences R_i^j by implementing feature selection algorithm on smaller datasets that are obtained by different ratio values. As a result, for each ratio value, r^j, the stability of the algorithm over q subsets of samples is given by:

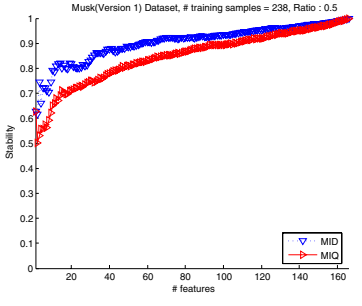
$$\frac{1}{q}\sum_{i=1}^{q} Sim^M\left(R_i, R_i^j\right) \tag{14}$$



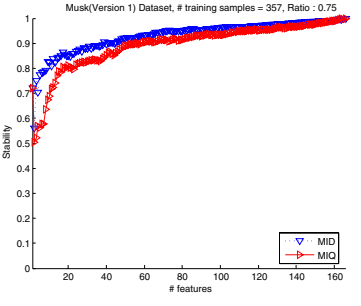
(a) Stability values of MID and MIQ while the number of selected features(x axis) varies when $r = 0.1$ for Musk (Version 1) dataset.



(b) Stability values of MID and MIQ while the number of selected features(x axis) varies when $r = 0.25$ for Musk (Version 1) dataset.



(c) Stability values of MID and MIQ while the number of selected features(x axis) varies when $r = 0.5$ for Musk (Version 1) dataset.



(d) Stability values of MID and MIQ while the number of selected features(x axis) varies when $r = 0.75$ for Musk (Version 1) dataset.

Fig. 3. Stability values of MID and MIQ for Musk (Version 1) dataset

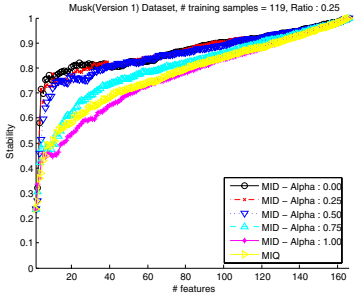
In our experiments we do stability and accuracy computation of feature selection algorithms as the size of available training samples decrease. In [7] stability computation is performed only for a single bootstrap sample which corresponds to approximately $r = 0.632$ ([15] p. 333).

The classifiers that are used in experiments are k-nearest neighbors classifiers with the k value of 3. The support vector machines were also used, however their accuracies did not show a significant difference, hence they are not given here.

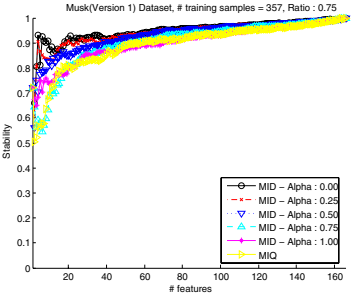
The stability comparison of all algorithms and the accuracy values of feature sequences computed on $D_{test,i}$ can be seen at the Results section.

5.3 Results

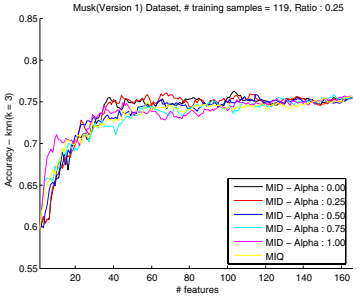
Stability of MID and MIQ : We first compared stability and accuracy of MID and MIQ on different datasets. In general MID is more stable than MIQ. The stability values of MID and MIQ for Musk (Version 1) dataset is shown



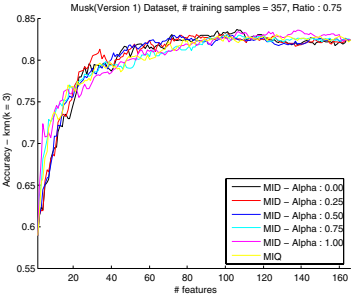
(a) Stability values while the number of selected features(x axis) varies when $r = 0.25$ for Musk (Version 1) dataset.



(b) Stability values while the number of selected features(x axis) varies when $r = 0.75$ for Musk (Version 1) dataset.



(c) Accuracy values computed with k-nn classifier($k = 3$), while the number of selected features(x axis) varies when $r = 0.25$ for Musk (Version 1) dataset.



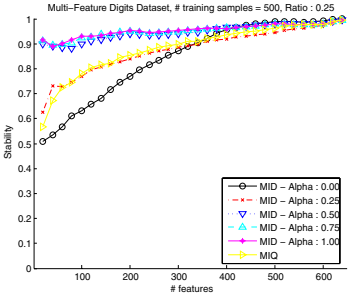
(d) Accuracy values computed with k-nn classifier($k = 3$), while the number of selected features(x axis) varies when $r = 0.75$ for Musk (Version 1) dataset.

Fig. 4. Stability and accuracy values for Musk (Version 1) dataset

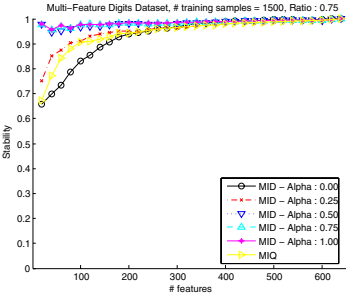
in Figure 3. As seen in the figure, MID is always more stable than MIQ, both for different values of r and the number of features selected, for this dataset.

Stability of MID, MIQ and MID_α : In order to compare stability and accuracy of MID, MIQ and MID_α we performed experiments on all the datasets. We demonstrate the mean stability and accuracy values for the Musk (Version 1) and Multi-features datasets in Figures 4 and 5. The stability and accuracy plots for the other datasets are omitted due to space restrictions but are available at [16].

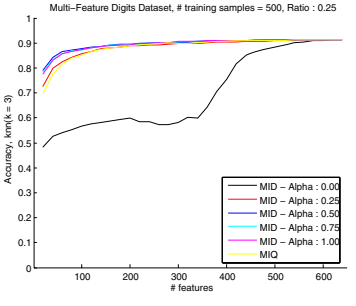
As seen in Figures 4 and 5, MID is again more stable than MIQ. However, it is not clear which value of α results in the most stable feature selection. In order to find the best value of parameter α in terms of stability and accuracy for each dataset, we propose comparing the mean ranks of each feature selection method MID_α for each number of selected features. For a certain number of selected features, we compute the rank of a method (smaller rank means better stability or accuracy),



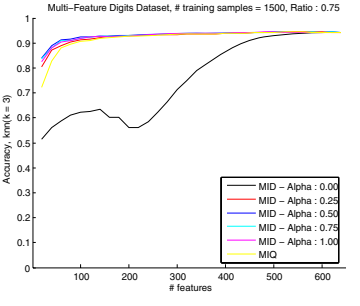
(a) Stability values while the number of selected features(x axis) varies when $r = 0.25$ for Multi-features dataset.



(b) Stability values while the number of selected features(x axis) varies when $r = 0.75$ for Multi-features dataset.



(c) Accuracy values computed with k-nn classifier($k = 3$), while the number of selected features(x axis) varies when $r = 0.25$ for Multi-features dataset.



(d) Accuracy values computed with k-nn classifier($k = 3$), while the number of selected features(x axis) varies when $r = 0.75$ for Multi-features dataset.

Fig. 5. Stability and accuracy values for Multi-features dataset

Table 3. Rank of α values and their standard deviation according to stability for Musk (Version 1) dataset

Ratio r	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
0.1	1.7 ∓ 0.9	1.8 ∓ 0.8	2.7 ∓ 0.8	3.9 ∓ 0.7	4.7 ∓ 0.8
0.25	1.9 ∓ 0.8	1.4 ∓ 0.6	2.7 ∓ 0.6	4 ∓ 0.4	4.8 ∓ 0.8
0.5	1.7 ∓ 0.7	1.5 ∓ 0.6	2.8 ∓ 0.5	4.1 ∓ 0.5	4.8 ∓ 0.7
0.75	1.4 ∓ 0.7	1.8 ∓ 0.6	2.9 ∓ 0.6	4.1 ∓ 0.6	4.7 ∓ 0.7

Table 4. Rank of α values and their standard deviation according to accuracy(knn = 3) for Musk (Version 1) dataset

Ratio r	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
0.1	4.2 ∓ 1.1	1.3 ∓ 0.6	1.9 ∓ 0.6	3.3 ∓ 0.8	4.1 ∓ 0.7
0.25	4.3 ∓ 1	1.1 ∓ 0.3	2 ∓ 0.4	3.3 ∓ 0.5	4.2 ∓ 0.8
0.5	4.5 ∓ 0.8	1.1 ∓ 0.4	1.9 ∓ 0.3	3.3 ∓ 0.7	4.1 ∓ 0.7
0.75	4.9 ∓ 0.4	1.1 ∓ 0.4	2 ∓ 0.5	3.2 ∓ 0.7	3.7 ∓ 0.6

then we average them for all the number of features. If two methods result in the same stability or accuracy, then we give them the same rank.

Table 3 shows the averages and standard deviations of stability ranks of MID_α methods and Table 4 shows the same ranks but for accuracy for Musk (Version 1) dataset.

Table 5. Rank of α values and their standart deviation according to stability for Multi-features dataset

Ratio r	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
0.1	3.3 ∓ 1.8	4.4 ∓ 0.6	2.4 ∓ 0.6	2.2 ∓ 0.8	2.7 ∓ 1.5
0.25	3.4 ∓ 1.9	4.5 ∓ 0.5	2.8 ∓ 0.6	2.2 ∓ 0.7	2.1 ∓ 1.3
0.5	3.3 ∓ 1.8	4.5 ∓ 0.5	3.2 ∓ 0.6	2.1 ∓ 0.8	1.9 ∓ 0.9
0.75	3.2 ∓ 1.8	4.5 ∓ 0.5	3.3 ∓ 0.6	2.1 ∓ 0.8	1.9 ∓ 1

Table 6. Rank of α values and their standart deviation according to accuracy(knn = 3) for Multi-features dataset

Ratio r	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
0.1	4.8 ∓ 0.7	3.9 ∓ 0.4	2.4 ∓ 0.9	1.8 ∓ 0.6	1.7 ∓ 0.8
0.25	4.7 ∓ 0.7	3.8 ∓ 0.7	1.9 ∓ 0.9	2.1 ∓ 0.6	2.1 ∓ 0.9
0.5	4.6 ∓ 1	3.6 ∓ 0.7	1.9 ∓ 0.9	2.1 ∓ 0.8	2.1 ∓ 1
0.75	4.7 ∓ 0.7	3.4 ∓ 1	1.8 ∓ 0.9	2 ∓ 0.7	2.2 ∓ 1

Table 7. Rank of α values and their standart deviation according to stability for all datasets when $r = 0.25$

Dataset	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
Ionosphere	3.5 ∓ 1.2	3 ∓ 0.9	3.4 ∓ 1.1	3 ∓ 1.9	1.4 ∓ 0.6
Sonar	2.2 ∓ 0.8	1.7 ∓ 0.8	2.6 ∓ 1.2	4.2 ∓ 0.8	4 ∓ 1.4
Parkinsons	2.5 ∓ 1.1	2.2 ∓ 1.1	3 ∓ 1.4	3.6 ∓ 1.3	2.7 ∓ 1.9
Musk(Version 1)	1.9 ∓ 0.8	1.4 ∓ 0.6	2.7 ∓ 0.6	4 ∓ 0.4	4.8 ∓ 0.8
Audio Genre	3.7 ∓ 1.1	4.5 ∓ 0.9	3.2 ∓ 0.7	1.8 ∓ 0.8	1.4 ∓ 0.5
Multi-features Digits	3.4 ∓ 1.9	4.5 ∓ 0.5	2.8 ∓ 0.6	2.2 ∓ 0.7	2.1 ∓ 1.3
Handwritten Digits	2.9 ∓ 1.5	4.4 ∓ 1	2.4 ∓ 1	2.3 ∓ 1.1	2 ∓ 1
Wine	3.7 ∓ 1.3	4.1 ∓ 1.4	2.5 ∓ 1	1.4 ∓ 0.5	1.8 ∓ 0.9

Table 8. Rank of α values and their standart deviation according to accuracy(knn = 3) for all datasets when $r = 0.25$

Dataset	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
Ionosphere	2.7 ∓ 1.9	3.2 ∓ 1.2	3.5 ∓ 1.2	3.1 ∓ 1.1	1.9 ∓ 1
Sonar	3.1 ∓ 1.9	1.8 ∓ 0.7	2.5 ∓ 1.2	3.7 ∓ 1	3.7 ∓ 1
Parkinsons	3.6 ∓ 1.6	2.2 ∓ 1.2	2.9 ∓ 1.4	3.2 ∓ 1	2.4 ∓ 1.3
Musk(Version 1)	4.3 ∓ 1	1.1 ∓ 0.3	2 ∓ 0.4	3.3 ∓ 0.5	4.2 ∓ 0.8
Audio Genre	4.8 ∓ 0.6	3.9 ∓ 0.7	2.9 ∓ 0.4	1.7 ∓ 0.6	1.4 ∓ 0.5
Multi-features Digits	4.7 ∓ 0.7	3.8 ∓ 0.7	1.9 ∓ 0.9	2.1 ∓ 0.6	2.1 ∓ 0.9
Handwritten Digits	4.4 ∓ 0.9	3.8 ∓ 1	2 ∓ 0.9	2 ∓ 0.9	1.7 ∓ 0.8
Wine	2.5 ∓ 1.4	4.3 ∓ 1.3	3.1 ∓ 1.3	1.8 ∓ 0.7	2.2 ∓ 0.7

Table 5 shows the averages and standard deviations of stability ranks of MID_α methods and Table 6 shows the same ranks but for accuracy for Multi-features dataset.

We summarize the stability and accuracy ranks of MID_α for all datasets in tables 7 and 8 respectively, when $r = 0.25$. According to these tables, $\alpha = 0.5$, i.e. MID is not necessarily the best choice in terms of either stability or accuracy. Different datasets favor different values of α for best stability and accuracy. While Ionosphere, audio genre, multi features, handwritten digits and wine datasets prefers $\alpha > 0.5$ for best stability, others perform better when $alpha < 0.5$.

6 Conclusion

In this paper, we first devised a method to evaluate the stability of a feature selection method as the number of dataset used for feature selection gets smaller. Then, using our stability evaluation method, on 8 different datasets, we showed that among the two feature selection criteria, MID and MIQ of the MRMR feature selection method, MID gives more stable results, while the accuracy of both criteria are comparable. We also showed theoretically, why MID is more stable than MIQ. Finally, we suggested an improvement on the MID criterion, MID_α, where the contribution of relevance and redundancy on feature selection is controlled through a parameter α. For different data sets, we evaluated the stability and accuracy performance of MID_α and observed that for each dataset the α that results in the best stability and accuracy could be different. Understanding what value of α is the best based on the characteristics of a dataset is one of the future works we are planning on.

Acknowledgments

The authors would like to thank Istanbul Technical University, Computer Engineering Department, Data Mining and Pattern Recognition Laboratory members Baris Senliol and Yusuf Yaslan for useful discussions.

References

1. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Feature and The Subset Selection Problem. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 121–129 (1994)
2. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
3. Ding, C., Peng, H.: Minimum Redundancy Feature Selection from Microarray Gene Expression Data. In: Proceedings of the Computational Systems Bioinformatics conference (CSB 2003), pp. 523–529 (2003)
4. Yu, L., Liu, H.: Efficient Feature Selection via Analysis of Relevance and Redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
5. Pepe, M.S., Etzioni, R., Feng, Z., et al.: Phases of Biomarker Development for Early Detection of Cancer. J. Natl. Cancer Inst. 93, 1054–1060 (2001)
6. Kalousis, A., Prados, J., Hilario, M.: Stability of Feature Selection Algorithms: A Study on High-Dimensional Spaces. Knowledge and Information Systems 12, 95–116 (2007)
7. Yu, L., Ding, C., Loscalzo, S.: Stable Feature Selection via Dense Feature Groups. In: Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2008), pp. 803–811 (2008)
8. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML 2008, Part II. LNCS (LNAI), vol. 5212, pp. 313–325. Springer, Heidelberg (2008)
9. Ding, I., Peng, H.C.: Minimum Redundancy Feature Selection from Microarray Gene Expression Data. In: Proc. Second IEEE Computational Systems Bioinformatics Conf., pp. 523–528 (2003)
10. Hungarian Algorithm by Alexander Melin, MATLAB CENTRAL Web Site, `http://www.mathworks.com/matlabcentral/fileexchange/11609`
11. Marsaglia, G.: Ratios of Normal Variables and Ratios of Sums of Uniform Variables. Journal of the American Statistical Association 60(309), 193–204 (1965)
12. UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
13. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing 10(5), 293–302 (2002)
14. Ding, I., Peng, H.C., Moerchen, F., Ultsch, A., Thies, M., Loehken, I.: Modelling Timbre Distance with Temporal Statistics From Polyphonic Music. IEEE Transactions on Speech and Audio Processing 14, 81–90 (2006)
15. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Cambridge (2004)
16. Gulgezen, G.: Stable and Accurate Feature Selection. M.Sc. Thesis, Istanbul Technical University, Computer Engineering Department (2009)