Stable Feature Selection via Dense Feature Groups

Lei Yu
Department of Computer Science
Binghamton University
Binghamton, NY 13902, USA
lyu@cs.binghamton.edu

Chris Ding
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019, USA
CHQDing@uta.edu

Steven Loscalzo
Department of Computer Science
Binghamton University
Binghamton, NY 13902, USA
sloscal1@binghamton.edu

ABSTRACT

Many feature selection algorithms have been proposed in the past focusing on improving classification accuracy. In this work, we point out the importance of stable feature selection for knowledge discovery from high-dimensional data, and identify two causes of instability of feature selection algorithms: selection of a minimum subset without redundant features and small sample size. We propose a general framework for stable feature selection which emphasizes both good generalization and stability of feature selection results. The framework identifies dense feature groups based on kernel density estimation and treats features in each dense group as a coherent entity for feature selection. An efficient algorithm DRAGS (Dense Relevant Attribute Group Selector) is developed under this framework. We also introduce a general measure for assessing the stability of feature selection algorithms. Our empirical study based on microarray data verifies that dense feature groups remain stable under random sample hold out, and the DRAGS algorithm is effective in identifying a set of feature groups which exhibit both high classification accuracy and stability.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-data mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Feature selection, stability, high-dimensional data, kernel density estimation, classification

1. INTRODUCTION

Feature selection, the problem of selecting a minimum subset of original features for best predictive accuracy, has attracted strong interest in the past several decades. A great variety of feature selection algorithms have been developed and proven to be effective in improving predictive accuracy for classification in many application domains [21]. The subtle issue of feature redundancy is also resolved by algorithms which minimize redundancy and maximize relevance among selected features for classification [2, 11, 22, 30].

However, a relatively neglected issue is the *stability* of selected feature sets, which remains an unresolved problem. This problem is particularly important for knowledge discovery from high-dimensional data, where the goal of knowledge discovery is often to identify features best explaining the differences between classes or subsets of samples from thousands of features. For example, in biological applications (e.g., microarrays, mass spectrometry), the primary goal of domain experts in conducting high-throughput experiments is often to detect leads for some biologically relevant marker genes or proteins, rather than building models for predicting diseases or phenotypes of novel samples [4, 23].

Although many feature selection algorithms are effective in selecting a subset of predictive features for sample class prediction, they are not necessarily reliable to identify candidate features for subsequent costly biological validation. One may be tempted to choose the set of features producing the best predictive accuracy as a starting point for validation. However, for the same data, many different subsets of features can result in the same or similarly good accuracy [12, 25]. The large number of predictive subsets and the disparity among them reveals the instability of feature selection algorithms. As a consequence, domain experts are unlikely to have the confidence to investigate any single subset of predictive features.

One cause of such instability is the classic goal of feature selection which aims to select a minimum subset of features necessary for constructing a classifier of best predictive accuracy [18, 19]. Many feature selection algorithms thus discard features which are relevant to the target concept but highly correlated to the selected ones. For the purpose of knowledge discovery from features, such minimum subset misses important knowledge about redundant features. Moreover, among a set of highly correlated features, different ones may be selected under different settings of a feature selection algorithm. The problem is usually severe for high-dimensional data with many highly correlated features.

Another cause of the instability of feature selection algorithms is the relatively small number of samples in high-dimensional data. Take microarray data for example, the typical number of features (genes) is thousands or tens of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-193-4/08/08 ...$5.00.

thousands, but the number of samples is often less than a hundred. For the same feature selection algorithm, a different subset of features may be selected each time when there is a slight variation in the training data. Such instability has been confirmed by our observations from experiments as well as recent work studying the stability of feature selection algorithms under training data variations [10, 17].

The two causes of instability are independent, and amplify the effect of each other on feature selection from data with many redundant features but limited samples. In order to provide domain experts with stable feature selection results, we have to overcome both causes of instability. In this paper, we propose a general framework for stable feature selection which aims to achieve not only *good classification accuracy* but also *stable feature selection results*.

Our framework is motivated by a key observation that in the sample space, the dense core regions (peak regions), measured by probabilistic density estimation, are stable with respect to sampling of the dimensions (samples). For example, a spherical Gaussian distribution in the 100-dimensional space will likely be a stable spherical Gaussian in any of the subspaces. The features near the core of the spherical Gaussian, viewed as a core group are likely to be stable under sampling, although exactly which feature is closest to the peak could vary. Another observation is that the features near the core region are highly correlated to each other, and thus should have similar relevance scores w.r.t. some class labels, assuming the class labels are locally consistent. Thus these features can be regarded as a single group in feature ranking. And we can pick any one of them in final classification. In this sense, the feature group is a stable entity.

The rest of the paper is organized as follows. In Section 2, we review previous work on feature selection, in contrast with our work. In Section 3, we introduce preliminaries on kernel density estimation. In Section 4, we describe in detail the proposed stable feature selection framework and the DRAGS algorithm under this framework. In Section 5, we propose a general measure of stability for feature selection results. Section 6 evaluates the effectiveness of the DRAGS algorithm in terms of both classification accuracy and stability based on microarray data sets. Section 7 provides a summary of this work and some future directions.

2. RELATED WORK

For many years, feature selection has been generally viewed as a problem of searching for an optimal subset of features guided by some evaluation measures. Various feature selection methods can broadly fall into the filter model and the wrapper model depending on their evaluation measures [18]. Filter methods use measures of intrinsic data characteristics [9, 21, 29], and wrapper methods rely on the performance of a predefined learning algorithm to evaluate the goodness of a subset of features [18]. For high-dimensional data, filter methods are often preferred due to their computational efficiency. As to search strategy, a simple way is to evaluate each feature independently and form a subset based on top-ranked features. Such univariate methods have been shown effective in some applications [13, 20]. However, they do not work well when features highly correlate or interact with each other. Various subset search methods evaluate features in a subset together and select a small subset of relevant but non-redundant features [2, 11, 22, 30]. They have shown improved classification accuracy over univariate methods. Another way is to weight all features together according to maximum margin. The margin can be defined either by the distance between a selected data point and its nearest neighbors in the same and different classes (as in ReliefF-based weighting) [6, **?**, 24] or by the distance between support vectors (as in SVM-based weighting) [14, 27]. An advantage of such methods is that optimal weights of features can be estimated by considering features together. A subset of top-ranked features can be selected based on a single pass of weighting features [6, **?**, 24] or a recursive feature elimination (RFE) procedure [14, 27].

All work discussed above only focuses on the generalization ability of feature selection methods, and pays little attention to their stability; methods were not deliberately designed to achieve stable results and hence not evaluated in terms of stability either. In contrast, our work addresses the two causes of instability of feature selection algorithms identified in Introduction. Another distinction is that our proposed framework identifies coherent feature groups and treats each group as a single entity during feature evaluation and subsequent learning tasks, while previous work treats each feature as an entity for evaluation and classification.

Clustering has been applied to feature selection, by clustering features and then selecting one (or a few) representative features from each cluster [3, 5, 15], or simultaneously clustering and selecting features [16], to form a final feature subset. Intuitively, clustering features can illuminate relationships among features and facilitate feature redundancy analysis; a feature is more likely to be redundant to some other features in the same cluster than features in other clusters. However, an optimal clustering result does not indicate that features in each cluster are coherent in terms of relevance and can be treated as a single entity. More importantly, existing feature clustering methods for feature selection do not consider the stability of feature groups, and therefore, are essentially different from our framework for stable feature selection based on dense feature groups.

Two recent papers have studied the stability issue of feature selection under small sample size, and compared a few existing feature selection algorithms [10, 17]. For each algorithm, they measured the stability of selected features when various random subsets of the same training data were used for feature selection. They both concluded that different algorithms which performed equally well for classification had a wide difference in terms of stability, and recommended to empirically choose the best feature selection algorithm according to both accuracy and stability measured by repeatedly sampling of the training data. Such procedure is computationally very costly, and is subject to ad hoc choice of a predefined pool of feature selection algorithms and classification algorithms used for evaluation. Moreover, the best outcome of such procedure is limited to the stability of existing feature selection algorithms which often suffer from the two causes of instability discussed in Introduction.

Significant effort is needed in order to have a comprehensive comparison of the stability of various existing feature selection algorithms which apply different evaluation measures and search strategies. Our work takes a paradigm shift from this direction, and is clearly different from previous work on stability study. To the best of our knowledge, our work is the first to propose a feature selection algorithm which directly provides stable feature selection results by addressing the two causes of instability.

3. PRELIMINARY

Kernel density estimation (known as Parzen window) is the most popular non-parametric method for estimating probabilistic density functions [26]. Given a data set of n data points $D = \{\mathrm{x}_i\}_{i=1}^n$ in the d-dimensional space R^d, a well-known multivariate kernel density estimator is given by

$$\hat{p}(\mathrm{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathrm{x} - \mathrm{x}_i}{h}\right), \tag{1}$$

where $\hat{p}(\mathrm{x})$ is an estimate of the unknown *pdf*, $K(\mathrm{x})$ is a radially symmetric, non-negative kernel function integrating to one, and h is a fixed bandwidth (window size).

In many applications of machine learning and pattern recognition, it is often useful to identify the modes of the underlying density $p(\mathrm{x})$, which are located at the zeros of the gradient $\nabla p(\mathrm{x}) = 0$. The mean shift procedure [7] is an elegant way to estimate the locations of these zeros without estimating the density. Given a data set D and a kernel function K as introduced before, the mean shift vector is defined by

$$\mathrm{m}_{h,K}(\mathrm{x}) = \frac{\sum_{i=1}^{n} \mathrm{x}_i K(\frac{\mathrm{x}-\mathrm{x}_i}{h})}{\sum_{i=1}^{n} K(\frac{\mathrm{x}-\mathrm{x}_i}{h})} - \mathrm{x}, \tag{2}$$

i.e, the difference between the weighted sample mean, using the kernel for weights, and x, the center of the kernel.

Let $\{\mathrm{y}_j\}_{j=1,2,...}$ denote the sequence of successive locations of the kernel K, where,

$$\mathrm{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathrm{x}_i K(\frac{\mathrm{y}_j-\mathrm{x}_i}{h})}{\sum_{i=1}^{n} K(\frac{\mathrm{y}_j-\mathrm{x}_i}{h})} \quad j = 1, 2, ... \tag{3}$$

is the weighted mean at y_j computed based on kernel K and y_1 is the center of the initial position of the kernel. Such iterative movement of the kernel along the direction defined by the mean shift vector can start with any data point $\mathrm{x} \in D$.

It is proven [7] that if a kernel K has a convex and monotonically decreasing profile, both sequences $\{\mathrm{y}_j\}_{j=1,2,...}$ and $\{\hat{p}(\mathrm{y}_j)\}_{j=1,2,...}$ converge, and $\{\hat{p}(\mathrm{y}_j)\}_{j=1,2,...}$ is monotonically increasing. In addition, the magnitude of each successive mean shift vector (derived from (2) and (3))

$$\mathrm{m}_{h,K}(\mathrm{y}_j) = \mathrm{y}_{j+1} - \mathrm{y}_j \tag{4}$$

converges to zero, and the gradient of the density estimate (1) computed at the stationary point y_c is zero $\nabla\hat{p}(\mathrm{y}_c) = 0$. Two simple kernels which satisfy the condition are the flat kernel and Gaussian kernel.

4. STABLE FEATURE SELECTION

Our proposed framework for stable feature selection identifies dense feature groups based on kernel density estimation, and treats features in each dense group as a coherent entity for feature selection.

4.1 Identification of Dense Feature Groups

Kernel density estimation operates on a set of data vectors $\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n$, defined by a d-dimensional feature space. In this work, we apply such method to estimate the density of a set of feature vectors $\mathrm{f}_1, \mathrm{f}_2, ..., \mathrm{f}_n$ in a data set. In order to do so, we need to transpose the data matrix representing the data set; original feature vectors become data vectors in the new feature space defined by the original data vectors.

Algorithm 1 DGF (Dense Group Finder)

Input: data $D = \{\mathrm{x}_i\}_{i=1}^n$, bandwidth h
Output: a number of dense feature groups $G_1, G_2, ..., G_m$
for $i = 1$ **to** n **do**
  Initialize $j = 1$, $\mathrm{y}_{i,j} = \mathrm{x}_i$
  repeat
    Compute $\mathrm{y}_{i,j+1}$ according to (3)
  until convergence
  Set stationary point $\mathrm{y}_{i,c} = \mathrm{y}_{i,j+1}$
  Merge $\mathrm{y}_{i,c}$ with its closest peak if their distance $< h$
end for
For every unique peak p_r, add x_i to G_r if $||\mathrm{p}_r - \mathrm{x}_i|| < h$
Optional: Eliminate feature groups of low density

We use the multivariate density estimator in (1) to evaluate the density of a feature; a feature with higher value of $\hat{p}(\mathrm{x})$ is denser than a feature with lower value.

Our proposed framework is motivated by a key observation that the dense core regions (peak regions), measured by probabilistic density estimation, are stable with respect to sampling of the dimensions. For example, in a spherical Gaussian distribution, data in each dimension follow the distribution

$$p(x_p) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x_p-\mu_p)^2/2\sigma^2},$$

where x_p is the coordinate in p-th dimension of a vector x. Thus, the total distribution in 100 dimensions is just

$$\prod_{p=1}^{100} p(x_p) = \frac{1}{(\sqrt{2\pi}\sigma)^{100}} e^{-||\mathrm{x}-\mu||^2/2\sigma^2}.$$

Clearly, taking off any 50 dimensions, the rest will still be a spherical Gaussian. The features near the core of the spherical Gaussian, viewed as a core group are likely to be stable under sampling, although exactly which feature is closest to the peak could vary.

In order to identify such dense feature groups, we need to group together dense features which are close to the same density peak. Based on the mean shift procedure, we propose the DGF (Dense Group Finder) algorithm. As shown in Algorithm 1, DGF first finds a number of unique density peaks in the data, and then decides dense feature groups based on density peaks. The main part of DGF is the iterative mean shift procedure for all n features, which has a time complexity of $O(\lambda n^2 d)$, where λ is the number of iterations for each mean shift procedure to converge, and d is the dimensionality in the transposed data (i.e., the number of samples in the original feature space). Normally, it only takes a few steps for each mean shift procedure to converge.

A difficulty in kernel density estimation is the choice of kernel bandwidth h. If h is very large, the whole data will have only one peak. On the other hand, if h is very small, every data point will be a peak. Fortunately, there is a nice way to estimate it from the K-Nearest Neighbors (KNN) point of view. For each data point x_i, we can find its KNNs, and compute the average distance from x_i to its KNNs. This average distance is a reasonable length which captures the local density near x_i. We can further compute the average of the average distance of KNNs for all data points to get a global average length. For a data set with n features, the possible values of K range from 1 to n. The smaller the

chosen K value (hence the smaller bandwidth h), the higher correlation features included in each dense group will have. Therefore, in order to find coherent dense feature groups, a reasonable K value should be sufficiently small but away from 1. Clearly, when K=1, the bandwidth will be zero, and every data point will be a peak.

The major difference of DGF from other mean-shift based clustering algorithms [8] lies in the last two steps after finding all the unique peaks. Clustering algorithms based on mode seeking aim to create continuity-based clusters among data points, and, therefore, they group all data points attracted to the same peak into one cluster of arbitrary shape. Each resulting cluster may contain data points with low density which are far way from the peak. Our goal is to identify dense feature groups, and therefore, DGF only includes features into a feature group if they are close to a unique peak. Feature groups of low density can be eliminated in an optional step. In our work, we eliminate a feature group G_r if the average distance of the associated density peak P_r to its KNNs is above the kernel bandwidth decided in the way described above.

4.2 Selection based on Dense Feature Groups

The maximum pair-wise distance among features in the same dense feature group identified by DGF is limited by the kernel bandwidth. Under a sufficiently small bandwidth $h > 0$, features in each dense feature group will be highly correlated to each other, and thus should have similar relevance scores with respect to some class labels, assuming the class labels are locally consistent. Thus these features can be regarded as a single group in relevance based ranking. And we can pick any one of them in final classification. Therefore, our general framework for stable feature selection is to first identify dense feature groups and then select relevant feature groups among dense feature groups. To decide the relevance of each dense group, the framework treats features in each dense group as a coherent entity.

We propose the DRAGS (Dense Relevant Attribute Group Selector) algorithm under this general framework. As shown in Algorithm 2, DRAGS first finds a number of dense feature groups based on DGF, and then evaluates the relevance of each feature group based on the average relevance of features in each group. DRAGS has the same time complexity as DGF if feature relevance is measured based on individual feature groups. DRAGS can be easily extended to consider interactions among feature groups when deciding group relevance under its general framework. In this work, since our investigative emphasis is on the effectiveness of dense feature groups for stable feature selection, we use the simple method of individual feature evaluation to determine the group relevance in DRAGS. As to relevance measures, various existing feature evaluation measures such as correlation, dependency, and distance [21] can be chosen depending on data characteristics. We use F-statistic, a commonly used statistical measure for identifying differentially expressed genes, as the relevance measure for experiments on microarray data sets.

For the sake of a simple model, like most other feature selection algorithms, DRAGS is able to provide a compact feature subset for classification by only selecting one representative feature from each dense and relevant feature group. Since features in each dense group are highly correlated, DRAGS naturally deals with the redundancy among relevant features. DRAGS also overcomes the two causes of instability discussed in Introduction. As to instability caused by eliminating redundant features, DRAGS keeps highly correlated features in a coherent feature group. Such coherent feature groups provide valuable knowledge about how relevant features are correlated. Features in all groups together provide a more comprehensive set of important features than any single subset of features selected by methods eliminating redundant features. As to instability caused by small sample size, DRAGS ensures the stability of feature groups identified from a small number of samples by evaluating the density of features and identifying dense feature groups.

Algorithm 2 DRAGS (Dense Relevant Attribute Group Selector)

Input: data D, bandwidth h, relevance measure $\Phi(\cdot)$
Output: selected relevant feature groups $G_1, G_2, ..., G_k$
Find dense feature groups $G_1, G_2, ..., G_m = \mathrm{DGF}(D, h)$
for $i = 1$ **to** m **do**
  Measure relevance $\Phi(G_i)$ based on average relevance of features in G_i
end for
Rank $G_1, G_2, ..., G_m$ according to $\Phi(G_i)$
Select top k most relevant groups (or based on a threshold)

5. STABILITY MEASURES

Measuring the stability of feature selection algorithms requires some similarity measures for two sets of feature selection results. Let $R_1 = \{G_i\}_{i=1}^{|R_1|}$ and $R_2 = \{G_j\}_{j=1}^{|R_2|}$ denote two sets of feature selection results, where each G_i and G_j represents a group of features. In a special case when each G_i and G_j only contains a single feature, R_1 and R_2 become two subsets of features. In such case, the similarity between R_1 and R_2 can simply be decided by

$$Sim_{ID}(R_1, R_2) = \frac{2|R_1 \cap R_2|}{|R_1| + |R_2|}, \tag{5}$$

where the subscript $_{ID}$ indicates that the similarity is decided by matching feature indices between the two subsets. Measures of similar forms have been used for assessing the stability of selected feature subsets in related papers [10, 17] discussed in Section 2. In this work, we develop a measure which extends existing similarity measures in two aspects. First, it is directly applicable to assess the similarity between two sets of feature groups in a general case. Second, it considers the similarity of feature values in addition to feature indices, which makes it informative when two feature subsets contain a large portion of different but highly correlated features. This general similarity measure for two sets of feature selection results is defined based on maximum weighted bipartite matching.

Given a bipartite graph $\mathbf{G} = (V, E)$, with vertex partition $V = V_1 \cup V_2$, and edge set $E = \{(u, v) | u \in V_1, v \in V_2\}$. $\mathbf{G}$ is called a weighted bipartite graph if every edge (u, v) is associated with a weight $w_{(u,v)}$, and a complete bipartite graph if every u in V_1 is adjacent to every v in V_2. A matching M in $\mathbf{G}$ is a subset of non-adjacent edges in E. 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 (e.g, the Hungarian algorithm) for finding an optimal solution.

Given two sets of feature selection results, $R_1 = \{G_i\}_{i=1}^{|R_1|}$ and $R_2 = \{G_j\}_{j=1}^{|R_2|}$, we model R_1 and R_2 together as a weighted complete bipartite graph $\mathbf{G} = (V, E)$, where $V = R_1 \cup R_2$, and $E = \{(G_i, G_j) | G_i \in R_1, G_j \in R_2\}$, and $w_{(G_i,G_j)}$ is determined by the similarity between a pair of feature groups G_i and G_j. The overall similarity between R_1 and R_2 is defined as:

$$Sim^M(R_1, R_2) = \frac{\sum_{(G_i,G_j)\in M} w_{(G_i,G_j)}}{|M|}, \qquad (6)$$

where M is a maximum matching in $\mathbf{G}$.

Depending on how to decide $w_{(G_i,G_j)}$, we differentiate two forms of Sim^M: Sim^M_{ID} and Sim^M_V, where the subscripts $_{ID}$ and $_V$ respectively indicate that each weight is decided based on feature indices or feature values. When G_i and G_j represent feature groups with more than one feature, for Sim^M_{ID}, each weight $w_{(G_i,G_j)}$ can be decided by the simple measure Sim_{ID} in (5); For Sim^M_V, each weight can be decided by the correlation coefficient between the centers or the most representative features of the two feature groups. In the special case when G_i and G_j represent individual features, for Sim^M_{ID}, since $w_{(G_i,G_j)} = 1$ for matching features and 0 otherwise, Sim^M_{ID} becomes Sim_{ID}; for Sim^M_V, each weight can be simply decided by the correlation coefficient between the two individual features. Therefore, the proposed similarity measure in (6) is a general measure for assessing the similarity between two sets of feature selection results.

Given the general similarity measure, we define *stability* of a feature selection algorithm as the average similarity of various sets of results produced by the same feature selection algorithm under training data variations. Let $Sim^M(R, R_i)$ denote the similarity between two sets of results R and R_i from the full set of samples and a subset of samples, respectively. Each subset of samples can be obtained by randomly sampling or bootstrapping the full set of samples. The stability of an algorithm over q subsets of samples is given by:

$$\overline{Sim}^M(R, R_i) = \frac{1}{q}\sum_{i=1}^{q} Sim^M(R, R_i)\ . \qquad (7)$$

It is worth to note that the stability can also be measured based on pair-wise similarity of results from different subsets of samples. We use formula (7) because it is more efficient to compute than pair-wise comparison. Moreover, it directly captures how different the result will be from the result obtained based on the full data, when some training samples are randomly removed.

6. EMPIRICAL STUDY

In this section, we empirically study the framework for stable feature selection based on dense feature groups. The study is conducted in two parts. In Section 6.2, we verify that dense feature groups are stable with respect to sample hold out. In Section 6.3, we verify that feature selection from dense feature groups according to group relevance produces feature groups which are both highly predictive and stable. Before we delve into experimental results and discussions, we first present the setup of various experiments.

6.1 Experimental Setup

Table 1: Summary of Microarray Data Sets

Data Set	# Genes	# Samples	# Classes
Colon	2000	62	2
Leukemia	7129	72	2
Lung	12533	181	2
Prostate	6034	102	2
Lymphoma	4026	62	3
SRBCT	2308	63	4

We experimented with six frequently studied public microarray data sets[1], characterized in Table 1. Following the original work on Colon data [1], for each data set, we normalized each feature vector so that the mean over its components is zero and the standard deviation is one. Note that because of the normalization, the Euclidean distance between two feature vectors x_i and x_j is related to r, the Pearson correlation between x_i and x_j: $|\mathrm{x}_i - \mathrm{x}_j|^2 = 2d(1-r)$, where d is the number of dimensions of the feature vectors. Due to such relationship, dense feature groups identified based on kernel function using Euclidean distance consist of features which are highly correlated to each other.

In order to evaluate the stability of dense feature groups under sample hold out, each data set was randomly partitioned into 3 folds, with each fold containing 1/3 of all the samples. Algorithm 1, DGF, was repeatedly applied to 2 out the 3 folds, while a different fold was hold out each time. This process was repeated 10 times for different partitions of the data set. Overall, a total of 10×3 different subsets of samples were used to generate different sets of feature groups by DGF. DGF was also applied to the full set of data in order to produce a reference set of feature groups R for $\overline{Sim}^M(R, R_i)$, the average $Sim^M(R, R_i)$ over 30 folds.

In order to evaluate the generalization ability and stability of Algorithm 2, DRAGS, each of the 30 subsets of samples in the previous study was used as the training set to select relevant feature groups from dense feature groups produced by DGF, and then train classifiers based on selected feature groups. The corresponding hold-out fold was used as the test set. One representative feature (the one with the highest average similarity to all other features in the group) from each selected group was used for both training and testing. Both sophisticated SVM (linear kernel) and simple KNN (K=1) classification algorithms (Weka's implementation [28]) were used to evaluate the generalization ability of the representative features of feature groups selected by DRAGS. The average predictive accuracy over the 30 folds was used as the measure for generalization ability. To confirm that the selected relevant dense feature groups remain stable, the stability of DRAGS was measured in the same setting as in the previous study for DGF, except that dense but irrelevant feature groups were excluded from the stability measurement.

As discussed in Section 2, feature clustering has been used for feature selection. For comparison purpose, we investigated whether simple K-means clustering can produce feature groups which are both stable and predictive. Without prior knowledge about the optimal number of clusters

[1] http://www.cs.binghamton.edu/~lyu/KDD08/data/

in each data set, the performance of K-means was evaluated under a wide range of K values. For each K value, K-means was repeated 50 times with random initial seeds, and the clustering result with minimum WSS (Within clusters Sum of Squared errors) was used for performance evaluation. First, we evaluated the stability of feature clusters from K-means under sample hold out. Under each K value, the stability of K feature clusters was evaluated in the same setting as DGF. Then, we evaluated the generalization ability and stability of the feature clusters selected based on relevance. Like existing work [16], a cluster was selected for classification if its representative feature was among the top k (k<K) according to relevance score. The rest of the procedure was the same as in evaluating DRAGS. In addition, we also tested the classification performance using representative features from all K clusters like in [3] and found the results (not included in the paper) were consistently inferior than those from the above approach.

Additionally, we evaluated the performance of a well-known feature selection algorithm for small sample classification, SVM-RFE (RFE in short) [14], under the same setting as DRAGS. RFE recursively eliminates features based on SVM. At each iteration, it trains an SVM classifier, ranks features according to some score function, and eliminates one or more features with the lowest scores. Since RFE is computationally intensive, as in [14], we chose to first eliminate half of the remaining features at each iteration and then switch to one feature at a time when only a small number of features (50 in these experiments) were left.

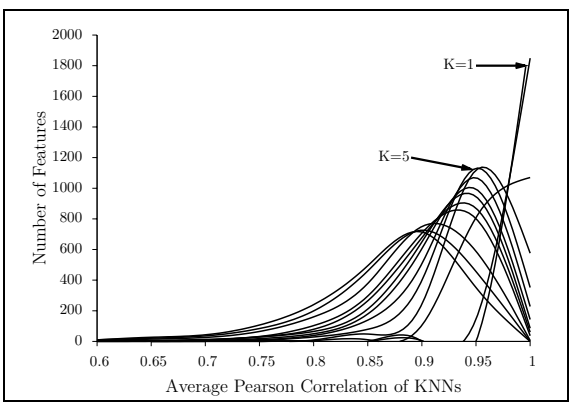

Figure 1: Various distributions of average Pearson correlation of K-Nearest Neighbors for all features, under K=1,2,...,10,15,20,25, for Colon data.

For DRAGS, we need to specify the number of nearest neighbors considered by each feature in order to determine the kernel bandwidth h for the mean shift procedure. Figure 1 depicts a series of distributions of average Pearson correlation of KNNs for all features under increasing values of K (from right to left) in Colon data set. We can see that the distributions are highly skewed and show little spread before K reaches 4, which indicates that the average Pearson correlation of KNNs does not adequately capture the heterogeneity of the underlying density distribution under those small values of K. Such information can be easily obtained based on pair-wise Pearson correlation among all features. We examined several data sets and observed very similar trend as in Figure 1. In order to find coherent dense feature groups, we prefer a K value which is small but able to capture the heterogeneity of the data. In our experiments, we uniformly set K=5 for all the data sets.

6.2 Evaluation of Dense Feature Groups

In this section, we evaluate the stability of dense feature groups produced by DGF under sample hold out based on the stability measure in (7), which has two forms $\overline{Sim}_V^M$ and $\overline{Sim}_{ID}^M$ depending on whether the similarity is measured based on feature values or feature indices. We also compare the stability of DGF with K-means algorithm under the setting described in Section 6.1. To compute $\overline{Sim}_V^M$, for both DGF and K-means, Pearson correlation of group centers is used to determine a maximum matching between two sets of feature groups R and R_i (R from the full data set and R_i from the ith random subset). To compute $\overline{Sim}_{ID}^M$, for DGF, features in each dense group are used to determine a maximum matching. For K-means, a maximum matching is determined in two ways: using all features in each cluster regardless of its size or 5 features closest to each cluster center (up to 5 if there are less than 5 features in the cluster). The higher stability value between the two is reported.

Figure 2 reports the stability values of DGF and K-means based on $\overline{Sim}_V^M$ and $\overline{Sim}_{ID}^M$ for each of the six microarray data sets used in our study. We can clearly observe from every data set that DGF is highly stable in terms of both measures when the top k (k=4, 6,..., 50) dense groups are evaluated. For all data sets except SRBCT, the stability score based on $\overline{Sim}_V^M$ is almost perfect for every k value, indicating Pearson correlation is almost 1 for all pairs of group centers under the best matching between two sets of feature groups. This observation verifies that density peaks in the sample space are highly stable with respect to sample hold out (even when 1/3 of the samples were removed in our experiments). The stability scores based on $\overline{Sim}_{ID}^M$ show the same trend, although they are less perfect than $\overline{Sim}_V^M$. Overall, more than 70% of the features in one dense feature group match with those in its matching group under the best matching for most k values, in five out of the six data sets. This further verifies that dense feature groups around density peaks are highly stable as well.

In contrast, K-means is much less stable than DRAGS in terms of both measures with only one exception (SRBCT, K=4). As the number of feature clusters increases, the stability of K-means degrades, that is, the resulting clusters become more sensitive to the variations of the dimensions (samples) included in computing the similarity between features. For all data sets, the $\overline{Sim}_{ID}^M$ scores are close to 0 for large numbers of clusters (e.g., k>20), which indicates almost no overlap between any pair of matching clusters, considering either all features in each cluster or several closest features to each cluster center. These observations suggest that grouping features without considering the density of feature groups is not effective for stable feature selection.

6.3 Evaluation of Feature Selection Results

We now evaluate the generalization ability and stability of selected feature groups by DRAGS. We also compare DRAGS with K-means based feature selection and RFE feature selection algorithm under the previously described set-

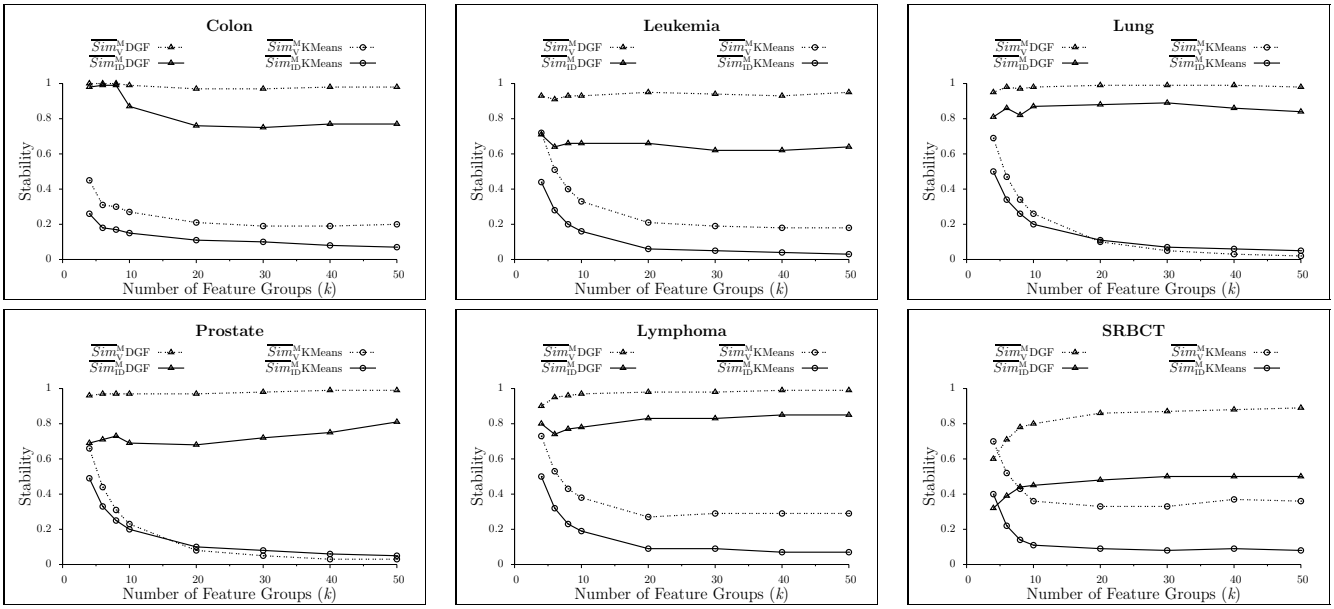

Figure 2: Stability of DGF and K-means according to $\overline{Sim}_V^M$ and $\overline{Sim}_{ID}^M$ measures for six data sets.

ting. Table 2 compares the average predictive accuracies (over 30 folds) for SVM and 1NN based on selection results from these three algorithms under a wide range of k, where k stands for the number of feature groups for DRAGS and K-means, and the number of features for RFE, respectively. The last value in each row is the average of accuracies across all the k values for each algorithm.

Between DRAGS and RFE, the accuracies resulted from DRAGS are significantly higher than those from RFE under all values of k for two data sets (Colon and SRBCT). For the other four data sets, DRAGS performs similar to RFE under large k values, but significantly higher when $k \leq 10$ (except Lymphoma). Such observations suggest that features in each dense group selected by DRAGS are coherent in terms of class discrimination, and therefore, good accuracy can be achieved by using representative features (one from each group) from only a few most relevant feature groups, rather than using a large subset of dozens of features like RFE. More importantly, DRAGS not only provides k features for classification, but also includes in its result features highly correlated to these k features. This is desirable for applications where the goal of knowledge discovery is to identify features best explaining the differences between classes. Comparing DRAGS with K-means based feature selection, the accuracies resulted from DRAGS are significantly higher than those from K-means under all values of k for SRBCT, and generally similar to those from K-means for the other five data sets.

At last, we evaluate the stability of DRAGS, K-means, and RFE. Figure 3 shows the stability of the three algorithms. We can clearly observe from all data sets that DRAGS remains highly stable in terms of both measures based on the top k relevant feature groups selected from dense feature groups. Therefore, we conclude that DRAGS can identify feature groups which together lead to good prediction of the class and are stable under sample hold out. K-means remains much less stable than DRAGS when the the top k relevant feature clusters among all K clusters are measured. For RFE, its stability values based on $\overline{Sim}_{ID}^M$ are consistently almost zero under any k value for all data sets, which shows that almost none of the features selected from a training fold matches with the set of features selected from the full data set. Its stability values based on $\overline{Sim}_V^M$ are higher due to the correlation between features selected based on a training fold and those selected based on the full data set, but RFE is overall much less stable than DRAGS. Such observations indicate that RFE is ineffective in providing stable results under training data variations, although it can select large subsets of features of good prediction.

7. CONCLUSION

In this paper, we have identified the importance of stable feature selection, and proposed a general feature selection framework for stable feature selection based on dense feature groups. We have also proposed a general measure of stability. Our empirical study based on various microarray data sets has verified that the proposed framework is effective for stable feature selection, and the DRAGS algorithm developed within this framework produces feature groups which together lead to good classification accuracy and are stable under sample hold out.

Because DRAGS limits the selection of relevant feature groups from dense feature groups identified by DGF, DRAGS may not necessarily include some of the most relevant features determined according to individual feature ranking in any of its selected feature groups, if those features are located in the sparse region of the data distribution. Some improvements to DRAGS can be studied in the future work. Another interesting future direction is to develop additional feature selection algorithms under the proposed framework, for example, by using other methods to evaluate the relevance of dense feature groups.

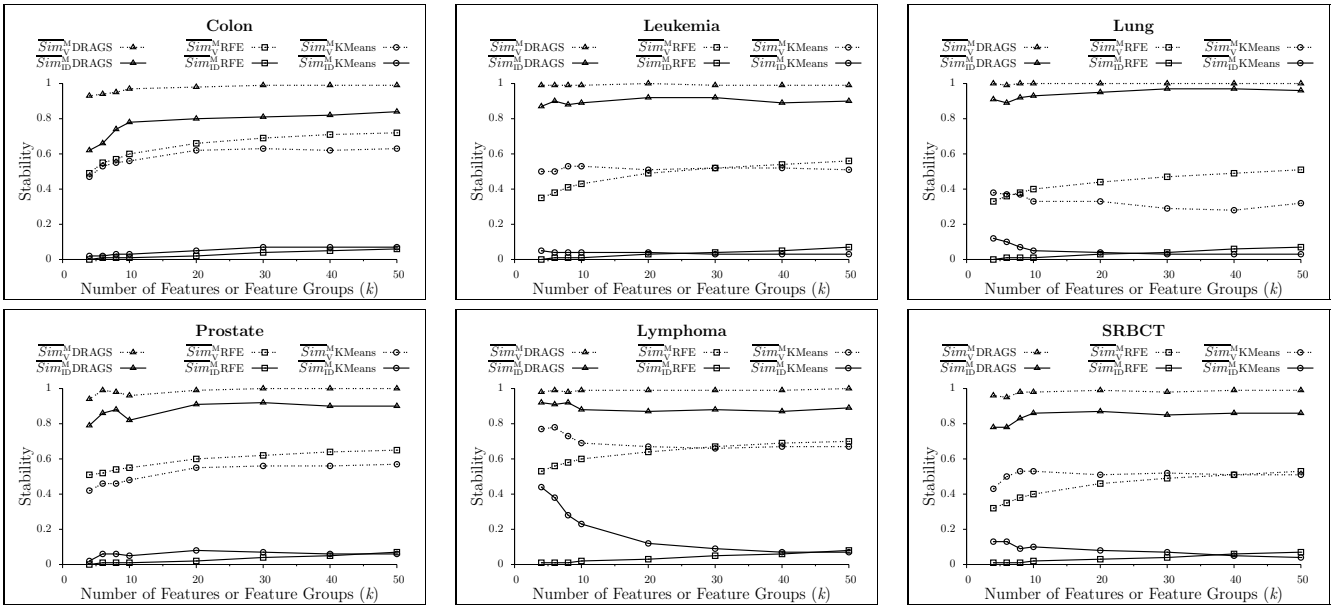

Figure 3: Stability of DRAGS, K-means, and RFE according to $\overline{Sim}_V^M$ and $\overline{Sim}_{ID}^M$ measures for six data sets.

8. ACKNOWLEDGMENTS

We would like to thank Yue Han for his effort in implementing the SVM-RFE algorithm and obtaining comparative results for it.

9. REFERENCES

[1] U. Alon, N. Barkai, D. A. Notterman, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, 96:6745–6750, 1999.

[2] A. Appice, M. Ceci, S. Rawles, and P. Flach. Redundant feature elimination for multi-class problems. In *Proceedings of the 21st International Conference on Machine learning*, pages 33–40, 2004.

[3] W. Au, K. C. C. Chan, A. K. C. Wong, and Y. Wang. Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):83–101, 2005.

[4] M. Berens, H. Liu, L. Parsons, L. Yu, and Z. Zhao. Fostering biological relevance in feature selection for microarray data. *IEEE Intelligent Systems*, 20(6):29–32, 2005.

[5] R. Butterworth, G. Piatetsky-Shapiro, and D. A. Simovici. On feature selection through clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 581–584, 2005.

[6] B. Cao, D. Shen, J. Sun, Q. Yang, and Z. Chen. Feature selection in a kernel space. In *Proceedings of the 24th International Conference on Machine learning*, pages 121–127, 2007.

[7] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

[8] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[9] M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial Intelligence*, 151(1-2):155–176, 2003.

[10] C. A. Davis, F. Gerick, V. Hintermair, C. C. Friedel, K. Fundel, R. Küffner, and R. Zimmer. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics*, 22:2356–2363, 2006.

[11] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics conference (CSB'03)*, pages 523–529, 2003.

[12] L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21:171–178, 2005.

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

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

[15] T. Hastie, R. Tibshirani, D. Botstein, and P. Brown. Supervised harvesting of expression trees. *Genome Biology*, 2:0003.1–0003.12, 2001.

[16] R. Jörnsten and B. Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19:1100–1109, 2003.

[17] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on

Table 2: Average Accuracies (%, with Standard Deviation ±) Produced by DRAGS, Kmeans, and RFE

Data		Method	k								Avg
			4	6	8	10	20	30	40	50	
Colon	SVM	DRAGS	**80.5±9.6**	**82.4±9.6**	**83.5±9.5**	**83.8±9.2**	**84.9±6.9**	**86.3±6.8**	**87.3±4.9**	**87.1±5.6**	**84.5**
		RFE	64.8±3.3	65.5±4.1	68.9±5.1	68.3±5.6	76.1±5.3	78.6±5.1	80.2±6.7	81.9±7.6	73.0
		Kmeans	77.6±10.7	79.3±8.7	81.7±7.7	82.7±7.0	84.2±6.3	84.2±5.8	81.9±8.2	82.2±7.8	81.8
	1NN	DRAGS	74.1±10.7	77±8.9	75.1±8.1	74.4±9.3	74.6±7.4	75.2±7.8	77.3±7.5	77.9±7.3	75.7
		RFE	59.9±6.2	64±5.8	67.2±3.4	66.9±5.9	73±4.6	74.8±5.8	75.9±5.1	80.3±4.9	70.3
		Kmeans	76.1±9.0	76.1±7.1	75.4±7.9	76.3±6.6	79.6±9.0	78.2±9.1	78.1±11.7	75.5±8.7	77.0
Leuk.	SVM	DRAGS	92.8±3.8	92.2±3.6	92.9±4	93.2±4.2	95±3.5	96.2±3	96.7±3.4	97.1±3.5	94.5
		RFE	78.2±5	85.5±3.7	87.3±5.5	89.9±4.9	95.3±2.9	95.5±2.7	96.9±0.9	97.5±0.9	90.8
		Kmeans	89.4±5.8	91.8±4.3	92.9±4.1	93.6±4.3	95.6±4.0	95.4±3.5	95.5±3.6	97.3±3.3	94.0
	1NN	DRAGS	93.5±4	92.1±4.1	90.6±4.8	90.8±4.3	92.4±3.6	92.4±5	94.3±3.5	95.6±4.5	92.7
		RFE	77.7±4.5	83.1±4.5	86.2±5	86.6±5.1	91.6±3	93.2±3.5	94.9±1.9	94.9±2.9	88.5
		Kmeans	90.2±5.4	90.6±6.2	91.3±5.7	93.3±4.9	93.8±4.3	94.1±4.1	93.7±3.7	95.1±4.2	92.8
Lung	SVM	DRAGS	98.5±1.9	99±1.3	98.9±1.3	98.7±1.8	98.8±1.6	98.9±1.4	99±1.3	99.1±1.1	98.9
		RFE	90.3±1.6	93.2±1.6	95.7±1.3	96.4±1.7	98.8±0.7	99.4±0.6	99.5±0.3	99.4±0.5	96.6
		Kmeans	94.9±3.2	96.0±2.8	97.1±2.5	97.4±2.5	97.6±2.6	97.8±2.1	98.3±1.8	98.4±2.0	97.2
	1NN	DRAGS	98.4±1.4	99±1.3	98.7±1.2	98.9±1.3	98.6±1.7	98.7±1.7	98.6±1.9	98.7±2	98.7
		RFE	91.9±3.1	93.6±2.2	95±1.7	95.2±2	97.7±1.1	97.6±0.9	98.4±0.5	98.4±0.8	96.0
		Kmeans	95.4±3.9	95.6±3.0	96.3±2.6	96.8±2.5	97.1±2.2	97.6±2.1	97.6±2.0	97.9±1.9	96.8
Pro.	SVM	DRAGS	86.6±4.6	88.6±4.9	90.5±5.5	89.8±5.9	89.1±6.4	89.9±5.4	91.3±4.6	91.7±3.9	89.7
		RFE	71.1±3.6	74.7±5	78.7±3.8	79.7±3.1	85.4±4.3	87.4±2.7	89.4±2.5	90.2±2.4	82.1
		Kmeans	83.1±6.8	85.1±7.3	85.3±7.2	85.7±7.0	88.7±6.0	88.0±5.9	87.4±5.4	88.1±5.2	86.5
	1NN	DRAGS	84.2±6.6	86.2±4.7	87.2±5.6	86.5±6.4	85.1±6.1	82.5±6.6	82.7±6.7	83.8±5.6	84.8
		RFE	64.3±4.8	70.7±4.6	74.7±3.2	77.4±4.2	79.8±4.3	83.2±5.1	83.9±3.6	84.6±2.8	77.3
		Kmeans	76.8±6.9	76.8±7.6	78.3±7.0	80±6.8	83.1±6.0	81.4±6.2	83.6±5.3	82.9±6.5	80.4
Lym.	SVM	DRAGS	82.4±8.5	86±11.3	94.3±10.7	94.8±10.3	99.2±1.8	98.3±3.9	97.5±4.8	98.6±4.5	93.9
		RFE	87.5±3.7	95.8±4	97.3±3.1	98±3.1	99±2.4	99.3±1.7	99.5±1.5	99.5±1.5	97.0
		Kmeans	87.9±9.5	91.9±9.4	94.4±7.6	94.4±7.7	97.3±3.6	97.6±3.7	98.2±2.9	98.4±2.8	95.0
	1NN	DRAGS	91.3±6	96.3±4.5	99±1.9	98.9±2.4	99.4±1.6	99±2.9	97.9±4.1	98.6±3.8	97.6
		RFE	95.2±3.8	97.7±2.9	97.3±3.7	98±3.1	98.8±2.5	98.8±2.2	99.2±1.9	99±2	98.0
		Kmeans	93.3±7.2	95.0±5.0	95.0±6.6	95.5±4.6	97.3±3.6	97.3±3.8	98.1±3.2	98.5±2.5	96.3
SRB.	SVM	DRAGS	**86.3±10.2**	**94.1±5.3**	**96.5±3.9**	**97±3.6**	**99.5±1.9**	**99.2±1.8**	**99.2±1.8**	**99.4±1.6**	**96.4**
		RFE	56.8±10.5	70.8±9.9	81.3±8.8	89.5±6.5	95.1±3.6	97±3.4	97.3±3.7	97.9±3.2	85.7
		Kmeans	61.4±15.3	72.7±16.1	79.0±15.6	85.4±11.3	90.7±6.3	93.0±5.5	93.9±5.9	93.1±6.8	83.7
	1NN	DRAGS	**92.2±6.3**	**96.7±4.2**	**96.7±4.2**	**97.3±3.7**	**99.5±1.5**	**99.8±0.9**	**99.4±1.6**	**99.2±1.8**	**97.6**
		RFE	76.5±10.3	80.3±7.5	86±7.1	88.3±8.5	92.4±5.1	92.5±6	93.3±5.2	94.6±4.5	88.0
		Kmeans	68.5±15.9	75.8±14.2	80.3±11.7	84.9±9.5	89.8±6.9	90.4±6.2	90.7±6.3	90.3±5.5	83.9

high-dimensional spaces. *Knowledge and Information Systems*, 12:95–116, 2007.

[18] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[19] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 284–292, 1996.

[20] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429–2437, 2004.

[21] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowledge and Data Engineering*, 17(3):1–12, 2005.

[22] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

[23] M. S. Pepe, R. Etzioni, Z. Feng, et al. Phases of biomarker development for early detection of cancer. *J Natl Cancer Inst*, 93:1054–1060, 2001.

[24] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, 53:23–69, 2003.

[25] R. L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, 19:1484–1491, 2003.

[26] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.

[27] L. Wang, J. Zhu, and H. Zou. Hybrid huberized support vector machines for microarray classification. In *Proceedings of the 24th International Conference on Machine learning*, pages 983 – 990, 2007.

[28] I. H. Witten and E. Frank. *Data Mining - Pracitcal Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.

[29] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608, 2001.

[30] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[31] Y. Zhang, C. Ding, and T. Li. A two-stage gene selection algorithm by combining reliefF and mRMR. *Proceedings of IEEE Conference of Bioinformatics and Bioengineering (BIBE2007)*, 2007.