Consensus Group Stable Feature Selection

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

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

ABSTRACT

Stability is an important yet under-addressed issue in feature selection from high-dimensional and small sample data. In this paper, we show that stability of feature selection has a strong dependency on sample size. We propose a novel framework for stable feature selection which first identifies consensus feature groups from subsampling of training samples, and then performs feature selection by treating each consensus feature group as a single entity. Experiments on both synthetic and real-world data sets show that an algorithm developed under this framework is effective at alleviating the problem of small sample size and leads to more stable feature selection results and comparable or better generalization performance than state-of-the-art feature selection algorithms. Synthetic data sets and algorithm source code are available at http://www.cs.binghamton.edu/∼lyu/KDD09/.

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, ensemble, high-dimensional data, small sample

1. INTRODUCTION

High-dimensional small-sample data is common in biological applications like gene expression microarrays [8] and proteomics mass spectrometry [20]. Classification on such data is challenging due to the two distinct data characteristics: high dimensionality and small sample size. Many feature selection algorithms have been developed with a focus on improving classification accuracy while reducing dimensionality for such data [3, 9, 10, 14, 15, 22]. The issues of feature relevance and redundancy have also been well studied [1, 5, 27]. 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 where feature selection is used as a knowledge discovery tool for identifying characteristic markers for the observed phenomena [19]. For example, in microarray data analysis, a feature selection algorithm may select largely different subsets of features (genes) under variations to the training data, although most of these subsets are as good as each other in terms of classification performance [11, 26]. Such instability dampens the confidence of domain experts in investigating any of the various subsets of selected features for biomarker identification.

In this paper, we demonstrate that stability of feature selection has a strong dependency on sample size. Moreover, we show that exploiting intrinsic feature groups in the underlying data distribution is effective at alleviating the effect of small sample size for high-dimensional data. Therefore, we propose a novel feature selection framework (as shown in Figure 1) which approximates intrinsic feature groups by a set of consensus feature groups and performs feature selection and classification in the transformed feature space described by consensus feature groups.

Our framework is motivated by a key observation that *intrinsic feature groups* (or groups of correlated features) commonly exist in high-dimensional data, and such groups are resistant to the variations of training samples. For example, genes normally function in co-regulated groups, and such intrinsic groups are independent to the set of observed microarray samples. Moreover, the set of intrinsic feature groups can be approximated by a set of consensus feature groups obtained from subsampling of the training samples. Another observation is that treating each feature group as a single entity and performing learning at the group level allows the ensemble effect of each feature group to offset the random relevance variation of its group members. Intuitively, it is less likely for a group of irrelevant features to exhibit the same trend of correlation to the class (hence, showing artificial relevance) than for each group member to gain some correlation to the class under random subsampling, unless all features in the group are perfectly correlated. Therefore, discriminating relevant groups from irrelevant ones based on group relevance is less prone to overfitting than detection of relevant features on small samples.

KDD'09, June 28–July 1, 2009, Paris, France.

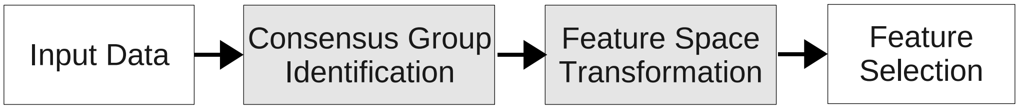


Figure 1: A framework of consensus group based feature selection.

As shown in Figure 1, there are two new issues in consensus group based feature selection: (1) identifying consensus feature groups from the given training data, and (2) representing each feature group by a single entity so that feature selection and classification can be performed on the transformed feature space. In our previous work [26], we developed an algorithm, DRAGS, which identifies dense feature groups in the sample space and uses a representative feature from each group in the subsequent feature selection and classification steps. The algorithm has shown some promising results w.r.t. the stability of the dense groups and the generalization ability of the selected features. However, there are two major limitations about DRAGS. First, DRAGS tries to identify dense feature groups in the sample space with dimensionality as high as dozens or a few hundreds (of samples) which makes density estimation difficult and unreliable. As a result, the feature groups found are not always stable under training data variations. Second, DRAGS faces the density vs. relevance dilemma - it limits the selection of relevant groups from dense groups for better stability of the selection results, however, it will miss some relevant features if those features are located in the relatively sparse regions. The new framework of consensus group based feature selection addresses these two issues.

The main contributions of this paper are: (i) conducting an in-depth study on the sample size dependency for the stability of feature selection; (ii) proposing a novel framework of consensus group based feature selection which alleviates the problem of small sample size; and (iii) developing a novel algorithm under this framework which overcomes the limitations of DRAGS. Experiments on both synthetic and real-world data sets show that the new algorithm leads to more stable feature selection results and comparable or better generalization performance than state-of-the-art feature selection algorithms DRAGS and SVM-RFE.

2. SAMPLE SIZE DEPENDENCY

High-dimensional data with small samples permits too large a hypothesis space yet too few constraints (samples), which makes learning on such data very difficult and prone to model overfitting. In order to find a probably approximately correct (PAC) hypothesis, PAC learning theory [12] gives a theoretic relationship between the number of samples needed in terms of the size of hypothesis space and the number of dimensions. For example, a binary data set with binary classes has a hypothesis space of size 2^{2^n} where n is the dimensionality. It would require $O(2^n)$ samples to learn a PAC hypothesis without any inductive bias [17].

Feature selection [7, 13] is one effective approach to reducing dimensionality - finding a subset of features from the original features. The reduction of dimensionality results in an exponential shrinkage of the hypothesis space, and hence reduces the chance of model overfitting and improves the generalization of classification algorithms [18]. However, feature selection itself is a challenging problem and receives increasing and intensified attention [16]. The shortage of samples in high-dimensional data increases the difficulty in finding relevant features, and reduces the stability of feature selection results under variations of training samples.

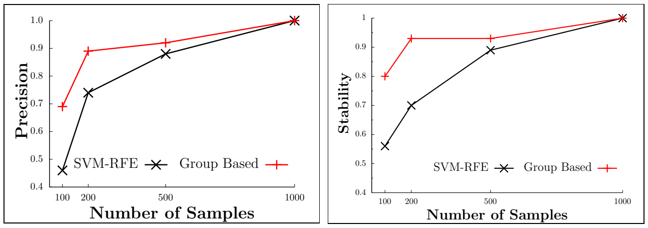
We next illustrate based on synthetic data that successful detection of relevant features and the stability of feature selection results can have a strong dependency on the sample size. The merit of using synthetic data for illustration is two-fold. First, it allows us to examine the sample size dependency using training sets with a wide range of sample sizes and other properties being equal; second, it provides us prior knowledge about truly relevant features.

The data set used here consists of 1000 training samples randomly drawn from the same distribution $P(X, Y)$. The feature set X consists of 1000 features, including 100 mutually independent features, $X_1, X_2, ..., X_{100}$, and a number of (10 ± 5) highly correlated features to each of these 100 features. Within each correlated group, the Pearson correlation of each feature pair is within (0.5,1), and the average pairwise correlation is below 0.75. The balanced binary class label Y is decided based on $X_1, X_2, ..., X_{10}$ using a linear function of equal weight to these 10 truly relevant features.

We study SVM-RFE [9], an algorithm well known for its excellent generalization performance on high-dimensional small-sample data. The main process of SVM-RFE is to recursively eliminate features based on SVM, using the coefficients of the optimal decision boundary to measure the relevance of each feature. At each iteration, it trains a linear SVM classifier, and eliminates one or more features with the lowest weights. We apply SVM-RFE on the above training set with sample size 1000, and its three randomly drawn subsets of training samples with decreasing sample sizes 500, 200, and 100, in order to observe the sample size dependency of SVM-RFE w.r.t. successful detection of relevant features and the stability of the selected feature subsets.

To evaluate the stability of SVM-RFE for a given training set, we can simulate training data variation by a resampling procedure like bootstrapping or N-fold cross-validation. We choose 10-fold cross-validation. For each training set, SVM-RFE is repeatedly applied to 9 out of the 10 folds, while a different fold is hold out each time. The stability of SVM-RFE is calculated based on the average pair-wise subset similarity of the top 10 features (the optimal number of features) selected over the 10 folds. To evaluate the effectiveness of SVM-RFE in detecting relevant features, the average precision of the top 10 features w.r.t the 10 truly relevant features over the 10 folds is also calculated.

To illustrate the effectiveness of feature selection based on intrinsic feature groups, we exploit the prior knowledge



(a) (b)

Figure 2: Precision (a) and stability (b) of the selected features by SVM-RFE and group-based SVM-RFE on various training sample sizes.

of existing feature groups in the synthetic data sets and replace each known feature group by its representative feature (the one closest to the group center). We then apply the SVM-RFE algorithm and the simple F-Statistic ranking to each transformed data set to selects the top 10 representative features, respectively. The group-based algorithms are evaluated in the same experimental setting as SVM-RFE.

Figure 2 shows the precision and stability of the selected top 10 features by SVM-RFE and the group-based SVM-RFE across different training sample sizes. The results of group-based F-Statistic ranking are almost the same as group-based SVM-RFE, and hence are not shown in the figures. Clearly, SVM-RFE shows a strong dependency on the training sample size w.r.t. successful detection of the truly relevant features as well as the stability of the selected features in this example. When the sample size reduces from 1000 to 100, both precision and stability curves drop sharply. In contrast, the group-based algorithm shows much less dependency on the sample size, especially when sample size is over 200. Moreover, the group-based algorithm consistently outperforms SVM-RFE. Such observations suggest that intrinsic feature groups are stable under training sample variations even at small sample size, and discriminating relevant features from irrelevant ones at the group level is more effective than at the feature level on small samples.

3. GROUP BASED FEATURE SELECTION

The study in the previous section illustrates the effectiveness of feature selection based on intrinsic feature groups in the underlying data distribution in an ideal situation. In practice, it is a challenging problem to identify intrinsic feature groups from a small training set due to the shortage of samples to observe feature correlation. In this section, we first review a previously proposed framework of *dense* group based feature selection and the DGF and DRAGS algorithms. We then describe the details of the new framework of *consensus* group based feature selection (as outlined in Figure 1) and a new algorithm under this framework.

3.1 Dense Group Based Feature Selection

The dense group based 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 [26].

Algorithm 1 DGF (Dense Group Finder)

```
Input: data $D = \{x_i\}_{i=1}^n$, kernel bandwidth $h$
Output: dense feature groups $G_1, G_2, \ldots, G_L$
for $i = 1$ to $n$ do
  Initialize $j = 1$, $y_{i,j} = x_i$
  repeat
    Compute $y_{i,j+1}$ according to (1)
  until convergence
  Set stationary point $y_{i,c} = y_{i,j+1}$ (make $y_{i,c}$ a peak $p_i$)
  Merge peak $p_i$ with its closest peak if their distance $< h$
end for
For every unique peak $p_r$, add $x_i$ to $G_r$ if $||p_r - x_i|| < h$
```

Given a training set D composed of n features and m samples, the data matrix is transposed such that original feature vectors become data points in the new feature space defined by the original samples. Algorithm 1, DGF (Dense Group Finder) was introduced in [26] as a means to locate dense feature groups from data. DGF works by employing kernel density estimation [24] and the iterative mean shift procedure [4] on each of the features in the sample space. When the mean shift process converges, nearby features are gathered into feature groups and returned by the algorithm.

The main part of DGF is the iterative mean shift procedure for all features, which locates a density peak by starting the mean at a given feature x_i and using other features in the local neighborhood (determined by a kernel bandwidth h) to shift the mean to a denser location. Specifically

$$y_{j+1} = \frac{\sum_{i=1}^n x_i K(\frac{y_j - x_i}{h})}{\sum_{i=1}^n K(\frac{y_j - x_i}{h})} \qquad j = 1, 2, ... \tag{1}$$

is used to determine the sequence of successive locations of the kernel K. The algorithm has a time complexity of $O(\lambda n^2 m)$, where λ is the number of iterations for each mean shift procedure to converge. Details of the algorithm and the choice of kernel function K and bandwidth are given in [26].

The groups found by DGF may or may not be relevant, and so these groups must be processed by a second algorithm DRAGS (Dense Relevant Attribute Group Selector) which relies on DGF to find dense feature groups and then evaluates and ranks the relevance of each feature group based on the average relevance (F-Statistic score) of features in each group. A representative feature (the one closest to the group center) from each selected top relevant group will be used for classification.

While the DGF and DRAGS algorithms have shown some promise w.r.t. the stability of the dense groups and the generalization performance of the selected features, the dense group based framework has two major limitations. In the first step, as mentioned in the Introduction, density estimation can be unreliable in high-dimensional spaces. Since the dimensionality of the feature space for estimating density peaks is as high as dozens or a few hundreds (i.e., the training sample size), the identified peaks are susceptible to variations of the dimensions (samples), and the stability of the identified dense groups suffer accordingly. Moreover, the overall sample size is still much smaller than the sample distribution, which can add to the instability of the groups

Algorithm 2 CGS (Consensus Group Stable Feature Selection

```
Input: data $D$, # of subsampling $t$, relevance measure $\Phi$
Output: selected relevant consensus feature groups $CG_1, CG_2, \ldots, CG_k$
// Identifying consensus groups
for $i = 1$ to $t$ do
    Construct bootstrapped training set $D_i$ from $D$
    Obtain dense feature groups by DGF $(D_i, h)$
end for
for every pair of features $X_i$ and $X_j \in D$
    Set $W_{i,j}$=frequency of $X_i$ and $X_j$ grouped together/$t$
end for
Create consensus groups $CG_1, CG_2, \ldots, CG_L$ by performing hierarchical clustering of all features based on $W_{i,j}$
// Feature selection based on consensus groups
for $i = 1$ to $l$ do
    Obtain a representative feature $X_i$ from $CG_i$
    Measure relevance $\Phi(X_i)$
end for
Rank $CG_1, CG_2, \ldots, CG_L$ according to $\Phi(X_i)$
Select top $k$ most relevant consensus groups
```

found under training sample variations. In the second step, the framework limits the selection of relevant groups from dense groups, and will miss some relevant features if those features are located in the relatively sparse regions. The new framework proposed next addresses these two issues.

3.2 Consensus Group Based Feature Selection

The consensus group based framework first approximates intrinsic feature groups by a set of consensus feature groups, and then performs feature selection in the transformed feature space described by consensus feature groups. We next describe each component in detail, and present a new algorithm CGS (Consensus Group Stable feature selection) which instantiates the proposed framework.

3.2.1 Identifying Consensus Groups

In practice, it is a challenging problem to identify intrinsic feature groups from a small training set due to the shortage of samples to observe feature correlation. Feature groups found on small samples can be suboptimal and instable under training sample variations. Our idea of approximating intrinsic feature groups by a set of consensus feature groups aggregated from multiple sets of feature groups originates from ensemble learning. It is well known that ensemble methods [2] for classification which aggregate the predictions of multiple classifiers can achieve better generalization than a single classifier, if the ensemble of classifiers are correct and diverse. Similar to the idea of ensemble classification, ensemble clustering methods [6, 23] have also be extensively studied, which aggregate clustering results from multiple clustering algorithms or from the same clustering algorithm under data manipulation. Although finding consensus feature groups can be considered as ensemble feature clustering, to the best of our knowledge, this is the first time that ensemble learning is applied to identify consensus feature groups for stable feature selection.

Similar to ensemble construction in classification and clustering, there are two essential steps in identifying consensus feature groups: Step (1), to create an ensemble of feature grouping results, and Step (2), to aggregate the ensemble into a single set of consensus feature groups. Algorithm 2, CGS , describes the key procedure for each of the two steps. In Step (1) CGS adopts the DGF algorithm introduced before as the base algorithm, and applies DGF on a number of (user-defined parameter t) bootstrapped training sets from a given training set D. The result of this step is an ensemble of feature groupings, $\{G_1^1, \ldots, G_{L_1}^1, \ldots, G_1^t, \ldots, G_{L_t}^t\}$, where G_j^i represents the j-th feature group formed in the ith DGF run. This straightforward step has time complexity $O(t\lambda n^2 m)$ as the base DGF algorithm has time complexity $O(\lambda n^2 m)$.

In Step (2), it is a non-trivial issue to aggregate a given ensemble of feature groupings $\{G_1^1, \ldots, G_{L_1}^1, \ldots, G_1^t, \ldots, G_{L_t}^t\}$ into a final set of consensus groups $\{CG_1, \ldots, CG_L\}$, where CG_i is a consensus group. This issue resembles the well-studied cluster ensemble problem - combining a given ensemble of clustering solutions into a final solution [6, 23]. Previously, Strehl and Ghosh [23] proposed two approaches, instance-based or cluster-based, to formulate the cluster ensemble problem. The instance-based approach models each instance as an entity and decides the similarity between each pair of instances based on how frequently they are clustered together among all clustering solutions. The cluster-based approach models each cluster in the ensemble as an entity, and decides the similarity between each pair of clusters based on the percentage of instances they share. Given the similarity matrix for all pairs of entities in either approach, a final clustering can be produced based on any hierarchical or graph clustering algorithm.

In this work, we chose the instance-based approach for the proposed CGS algorithm for two reasons. First, this approach is more efficient than the cluster-based approach w.r.t. both computation and space as the number of entities in the instance-based approach (i.e., the number of features) is often much smaller than the number of entities in the cluster-based approach (i.e., the total number of feature groups $\sum_{i=1}^{t} L_i$ in the ensemble) under the experimental settings described in the following section. Second, our preliminary evaluation of both approaches shows that the consensus groups formed by the instance-based approach are consistently more stable than the cluster-based approach. Once $W_{i,j}$ for every feature pairs is computed, the CGS algorithm applies agglomerative hierarchical clustering to group features into a final set of consensus feature groups. To reduce the effect of outliers, we use average linkage in deciding the similarity between two groups to be merged. The merging process continues as long as the two feature groups to be merged has a similarity value > 0.5, indicating, on average, the feature pairs in the resulting group are also grouped together by a majority of the DGF runs. The use of majority voting provides a natural stopping criterion for deciding the final number of feature groups. The time complexity for Step (2) is $O(n^2 t + n^2 logn)$.

3.2.2 Feature Selection based on Consensus Groups

Consensus groups found by CGS can still be comprised of irrelevant features, so, CGS continues to identify relevant groups from the consensus feature groups. CGS works by first forming a representative feature for each consensus feature group, and then evaluates the relevance of each feature group based on its representative feature. In our implementation, we use the feature closest to the group center to represent the group. Different feature selection algorithms and relevance measures can be adopted in the same framework to select relevant feature groups since each group has been represented by a single entity. In this work, since our investigative emphasis is on the effectiveness of consensus feature groups for stable feature selection, we use the simple method of individual feature evaluation based on F-Statistic to determine the group relevance in CGS, as we did for DRAGS in [26]. However, there are two key differences between CGS and DRAGS. First, CGS relies on consensus feature groups, while DRAGS relies on dense feature groups. Second, CGS considers all consensus groups during the relevance selection step, while DRAGS limits the selection of relevant groups to the top dense groups. Therefore, CGS addresses two key limitations of DRAGS discussed previously.

4. EMPIRICAL STUDY

In this section, we empirically study the framework of stable feature selection based on consensus feature groups. Section 4.1 introduces stability measures, Section 4.2 describes the data sets used and experimental procedures, and Section 4.3 presents results and discussion.

4.1 Stability Measures

Evaluating the stability of feature selection algorithms requires some similarity measures for two sets of feature selection results. In our previous work [26], we introduced a general similarity measure for two sets of feature selection results $R_1 = \{G_i\}_{i=1}^{|R_1|}$ and $R_2 = \{G_j\}_{j=1}^{|R_2|}$, where R_1 and R_2 can be either two sets of features or two sets of feature groups. R_1 and R_2 together are modeled as a weighted bipartite graph $\mathbf{G} = (V, E)$, with vertex partition $V = R_1 \cup R_2$, and edge set $E = \{(G_i, G_j) | G_i \in R_1, G_j \in R_2\}$, and weight $w_{(G_i,G_j)}$ associated with each pair $(G_i,\ G_j)$. The overall similarity between R_1 and R_2 is defined as:

$$Sim(R_1, R_2) = \frac{\sum_{(G_i,G_j)\in M} w_{(G_i,G_j)}}{|M|}, \qquad (2)$$

where M is a maximum matching in $\mathbf{G}$ (i.e., a subset of non-adjacent edges in E with largest sum of weights).

Depending on how to decide $w_{(G_i,G_j)}$, the measure has two forms: Sim_{ID} and Sim_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, for Sim_{ID}, each weight $w_{(G_i,G_j)}$ is decided by the overlap between the two feature groups,

$$w_{(G_i,G_j)} = \frac{2|G_i \cap G_j|}{|G_1| + |G_2|}. \qquad (3)$$

For Sim_V, each weight is decided by the Pearson correlation coefficient between the centers of the two feature groups. In the special case when G_i and G_j only contain one feature, for Sim_{ID}, $w_{(G_i,G_j)} = 1$ for matching features and 0 otherwise; for Sim_V, each weight can be simply decided by the correlation coefficient between the two individual features.

Given the similarity measure, the stability of a feature selection algorithm is then measured as the average similarity of various feature selection results produced by the same algorithm under training data variations. In [26], we used the feature selection result from a full training set as a reference to compare various results under subsampling of the full training set. Although the procedure is more efficient than pair-wise comparison among various results, the evaluation result can be biased since the individual difference between two sets of results can be greater than their difference to a reference set. We use pair-wise similarity comparison in stability calculation in this paper.

4.2 Experimental Setup

We perform our study on both synthetic data sets and real-world data sets. Besides the synthetic data set used in Section 2, we also create another data set with higher dimensionality and larger feature groups. A summary of these data sets is provided in Table 1. To assure comparable results, we follow the same procedure in generating both data sets as described in Section 2. For each data set, four different training sample sizes (100, 200, 500, and 1000) will be used to study the sample size dependency of the group-based algorithms as in Section 2. For real-world data, we use six frequently studied public microarray data sets characterized in Table 2.

Table 1: Summary of Synthetic Data Sets.

Data Set	Features	Feature Groups	Truly Rel. Feat.
D_{1k}	1000	100 (size 10 ± 5)	10
D_{5k}	5000	250 (size 20 ± 5)	10

Table 2: Summary of Microarray Data Sets

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

To empirically evaluate the stability and accuracy of SVM-RFE, DRAGS, and CGS on a given data set, we apply the 10 fold cross-validation procedure. For each microrray data set described above, each feature selection algorithm is repeatedly applied to 9 out of the 10 folds, while a different fold is hold out each time. Different stability measures are calculated. In addition, a classifier is trained based on the selected features from the same training set and tested on the corresponding hold-out fold. The CV accuracies of linear SVM and KNN classification algorithms are calculated. The above process is repeated 10 times for different random partitions of the data set, and the results are averaged. For each of the two synthetic data sets, we follow the same 10×10-fold CV procedure above with two changes. First, an independent test set of 500 samples randomly generated from the same distribution as the training set is used in replacement of the hold-out fold. Second, in addition to stability and accuracy measures, we also measure the precision w.r.t. truly relevant features during each iteration of the 10×10 CV and obtained the average values. For each performance measure, two-sample paired t-tests between the best performing algorithm and the other two algorithms is used to decide the statistical significance of the difference between the two average values over the 10 random trials.

As to algorithm settings, for SVM-RFE, we eliminate 10 percent of the remaining features at each iteration. We use

Weka's implementation [25] of SVM (linear kernel, default C parameter) and KNN (K=1). For DRAGS, the selection of relevant groups is limited to the top 50 dense feature groups. The kernel bandwidth parameter h for the base DGF algorithm is set to be the average of the average distance to its K-nearest neighbors for all features. As discussed in [26], a reasonable K value should be sufficiently small in order to capture the heterogeneity of the data. In our experiments, for each synthetic data, the K value is set to be the average group size. For each microarray data set, the K value is chosen from 3 to 5 based on the stability of DGF through cross-validation. For CGS , the number of subsampling t is set to be 10.

4.3 Results and Discussion

4.3.1 Synthetic Data

Figure 3 compares SVM-RFE, DRAGS, and CGS algorithms by various performance measures on the two synthetic data sets D_{1k} and D_{5k} under increasing training sample size. For SVM-RFE, the same trends as seen in Figure 2, Section 2 can be observed here on both data sets. SVM-RFE shows a strong dependency on the sample size w.r.t. successful detection of the truly relevant features (as shown in the precision figures in the left column) as well as the stability of the selected features (the Sim_{ID} figures in the middle column). The SVM accuracies based on the selected features (right column) are consistent with the precision values.

CGS also shows the same trends as the group-based algorithms which perform feature selection based on representative features of the intrinsic groups in Section 2. Our initial experimental results (not included due to space limit) showed that the consensus groups identified by the ensemble version of the DGF algorithm approximate the intrinsic groups very well on these synthetic data sets even when the sample size is small. As a consequence, CGS which performs feature selection based on representative features of the consensus groups shows a much less dependency on the sample size than SVM-RFE. A close look at Figure 3 shows that the performance of CGS at sample size 200 is usually as good as SVM-RFE at over twice the sample size. Such observation indicates that consensus group based feature selection is an alternative way of improving the stability of feature selection instead of increasing the sample size. It is worthy to note that in many applications, increasing the number of training samples could be impractical or very costly. For example, in gene expression microarray data, each sample is from the tissue of a cancer patient, which is usually hard to obtain and costly to perform experiments on.

The inferior performance of DRAGS may appear surprising at the first look. Such performance is due to the the fact that for each data set, different feature groups have similar density, and the ratio of relevant to irrelevant groups is low (1/9 for D_{1k} and 1/24 for D_{5k}). Therefore, the probability that a relevant group happens to be among the top 50 dense groups and considered by DRAGS is low. If DRAGS allowed all groups found by DGF to be considered for relevant group selection, its performance would be better on these data sets since the groups found by DGF are reasonably close to the true feature groups. However, for real-world data which could have heterogenous density among various groups, the dense group based framework faces the dilemma of the tradeoff between density v.s. relevance. Allowing DRAGS to select features from low density groups may increase the selection accuracy but the low density groups are sensitive to training data variations. The consensus group based framework proposed in this work avoids the above dilemma; it does not limit the selection from dense groups, and it improves the stability of the resulting feature groups based on the ensemble mechanism.

4.3.2 Microarray Data

Figures 4 and 5 compare SVM-RFE, DRAGS, and CGS by various performance measures on the six microarray data sets used in this study. Figures in the left column compare the stability of CGS and DRAGS w.r.t. the similarity of the selected features groups. CGS shows significantly better stability than DRAGS for all six data sets except Leukemia and Lung. This verifies the effectiveness of the ensemble mechanism of CGS at stabilizing the feature groups produced by the DGF algorithm. Figures in the middle column compare the stability of CGS , DRAGS, and SVM-RFE, w.r.t. the similarity of the selected features. CGS is significantly better than DRAGS for all six data sets except Leukemia. Overall, the stability of CGS is the best among all three algorithm in comparison. Figures in the right column compare the SVM accuracy of the three algorithms. CGS in general results in significantly higher accuracy than DRAGS and SVM-RFE on two data sets, Colon and SRBCT, and comparable results in the other data sets. For all data sets, the stability trends w.r.t. Sim_V measure (in Section 4.1) are consistent with those w.r.t Sim_{ID}, and the accuracy trends from KNN are in general similar to SVM. Due to space limit, curves for Sim_V and KNN accuracy are not reported.

Although CGS is computationally more costly than DRAGS and SVM-RFE, the payoff of significantly improved stability makes CGS a valuable tool for biologists who seek to identify not only highly predictive features but also stable feature groups. Such feature groups provide valuable insights about how relevant features are correlated, and may suggest high-potential candidates for biomarker detection.

5. RELATED WORK

There exist very limited studies on the stability of feature selection algorithms. An early work in this direction was done by Kalousis et al. (2007) . Their work raised the issue of feature selection stability and compared the stability of a number of conventional feature selection algorithms under training data variation based on three stability measures on high-dimensional data. More recently, two approaches were proposed to explicitly achieve stable feature selection without sacrificing classification accuracy: the dense group based feature selection in our previous work [26], and ensemble feature selection [21]. In the later, Saeys, et al. studied ensemble feature selection which aggregates the feature selection results from a conventional feature selection algorithm such as SVM-RFE repeatedly applied on different bootstrapped samples of the same training set. Their results show that the stability of ensemble SVM-RFE does not improve significantly from single run of SVM-RFE. Our ensemble approach in the proposed feature selection framework is different as it applies the idea of ensemble learning to identify consensus feature groups instead of consensus feature rankings or feature subsets. We also evaluated ensemble SVM-RFE and observed a similar trend as in [21] in our initial study.

6. CONCLUSION

In this paper, we have studied the sample size dependency for stability of feature selection. We have proposed a novel consensus group based framework for stable feature selection. Experiments on both synthetic and real-world data sets show that the CGS algorithm is effective at alleviating the problem of small sample size, and the algorithm in general leads to more stable feature selection results and comparable or better generalization performance than two state-of-the-art feature selection algorithms, DRAGS and SVM-RFE. Future work is planned to investigate different ensemble methods for identifying consensus feature groups, for example, different ways of generating and aggregating ensemble based on DGF algorithm or other robust clustering algorithms.

7. ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their helpful comments. C. Ding is partially supported by NSF-CCF-0830780 and NSF-DMS-0844497.

8. REFERENCES

[1] 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.

[2] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–142, 1999.

[3] X. Chen and J. C. Joeng. Minimum reference set based feature selection for small sample classifications. In *Proceedings of the 24th international conference on Machine learning*, pages 153 – 160, 2007.

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

[5] 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.

[6] X. Z. Fern and C. Brodley. Random projection for high-dimensional data clustering: a cluster ensemble approach. In *Proceedings of the twentieth International Conference on Machine Learning*, pages 186–193, 2003.

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

[8] T. R. Golub, D. K. Slonim, P. Tamayo, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

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

[10] K. Jong, J. Mary, A. Cornuejols, E. Marchiori, and M. Sebag. Ensemble feature ranking. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 267–278, 2004.

[11] 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.

[12] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

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

[14] 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.

[15] H. Liu, J. Li, and L. Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13:51–60, 2002.

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

[17] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[18] A. Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 404–412, 1998.

[19] 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.

[20] E. F. Petricoin, A. M. Ardekani, B. A. Hitt, et al. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359:572–577, 2002.

[21] Y. Saeys, T. Abeel, and Y. V. Peer. Robust feature selection using ensemble feature selection techniques. In *Proceedings of the ECML Confernce*, pages 313–325, 2008.

[22] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 823–830, 2007.

[23] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

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

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

[26] L. Yu, C. Ding, and S. Loscalzo. Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD-08)*, pages 803–811, 2008.

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

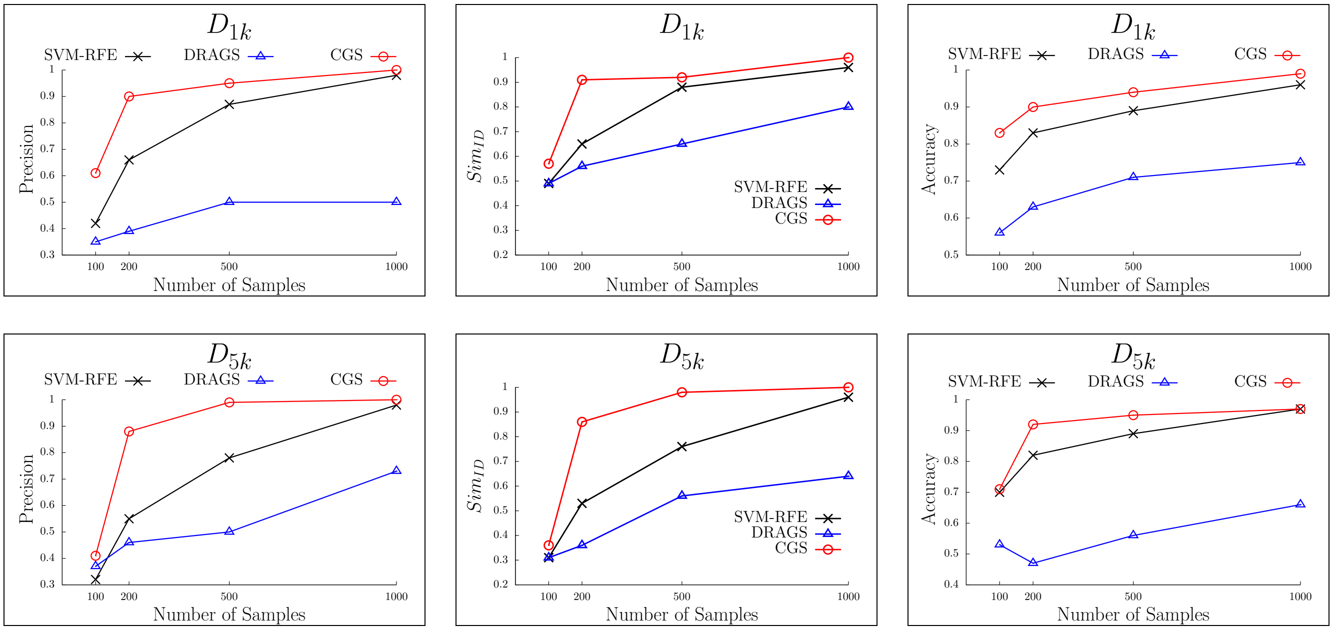


Figure 3: Comparison of SVM-RFE, DRAGS, and CGS on *synthetic* data sets. Figures in the left, middle, and right columns respectively show the precision w.r.t. the truly relevant features, the stability w.r.t. Sim_{ID} of the selected representative features, and the SVM classification accuracy, for the top 10 selected representative features under increasing sample size.

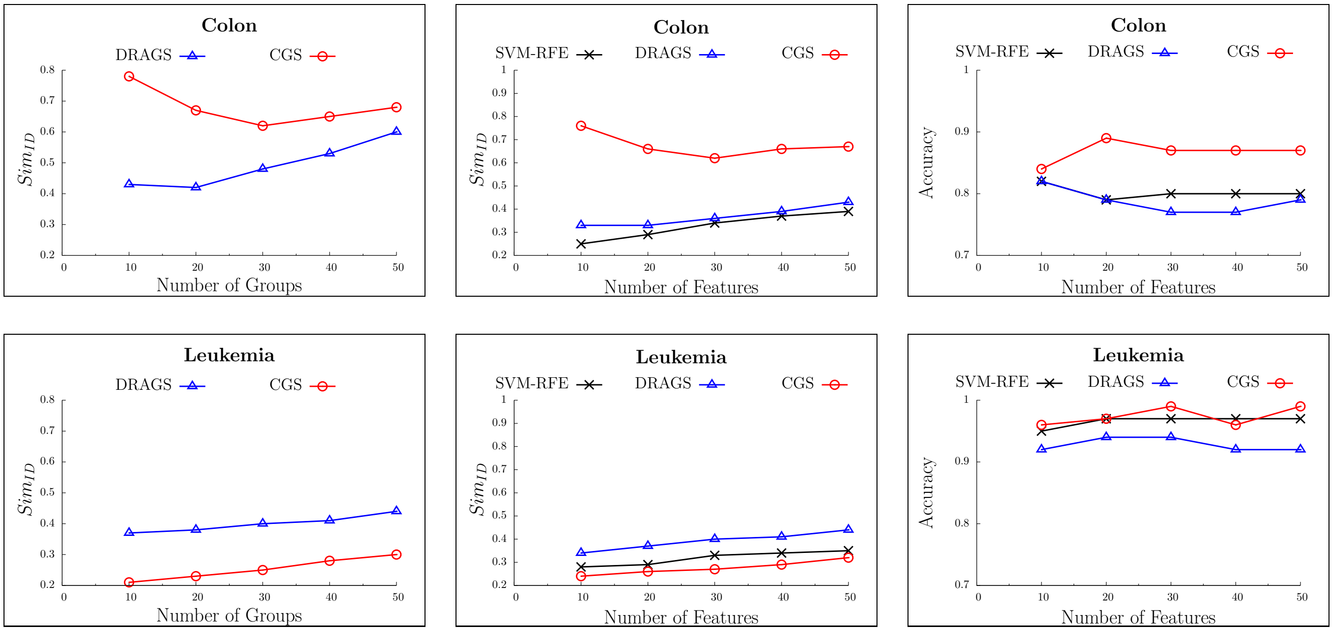


Figure 4: Comparison of SVM-RFE, DRAGS, and CGS on *Colon* and *Leukemia* microarray data sets. Figures in the left, middle, and right columns respectively show the stability w.r.t. Sim_{ID} of the selected feature groups, the stability w.r.t. Sim_{ID} of the selected representative features, and the SVM classification accuracy, for various numbers of selected features (or groups).

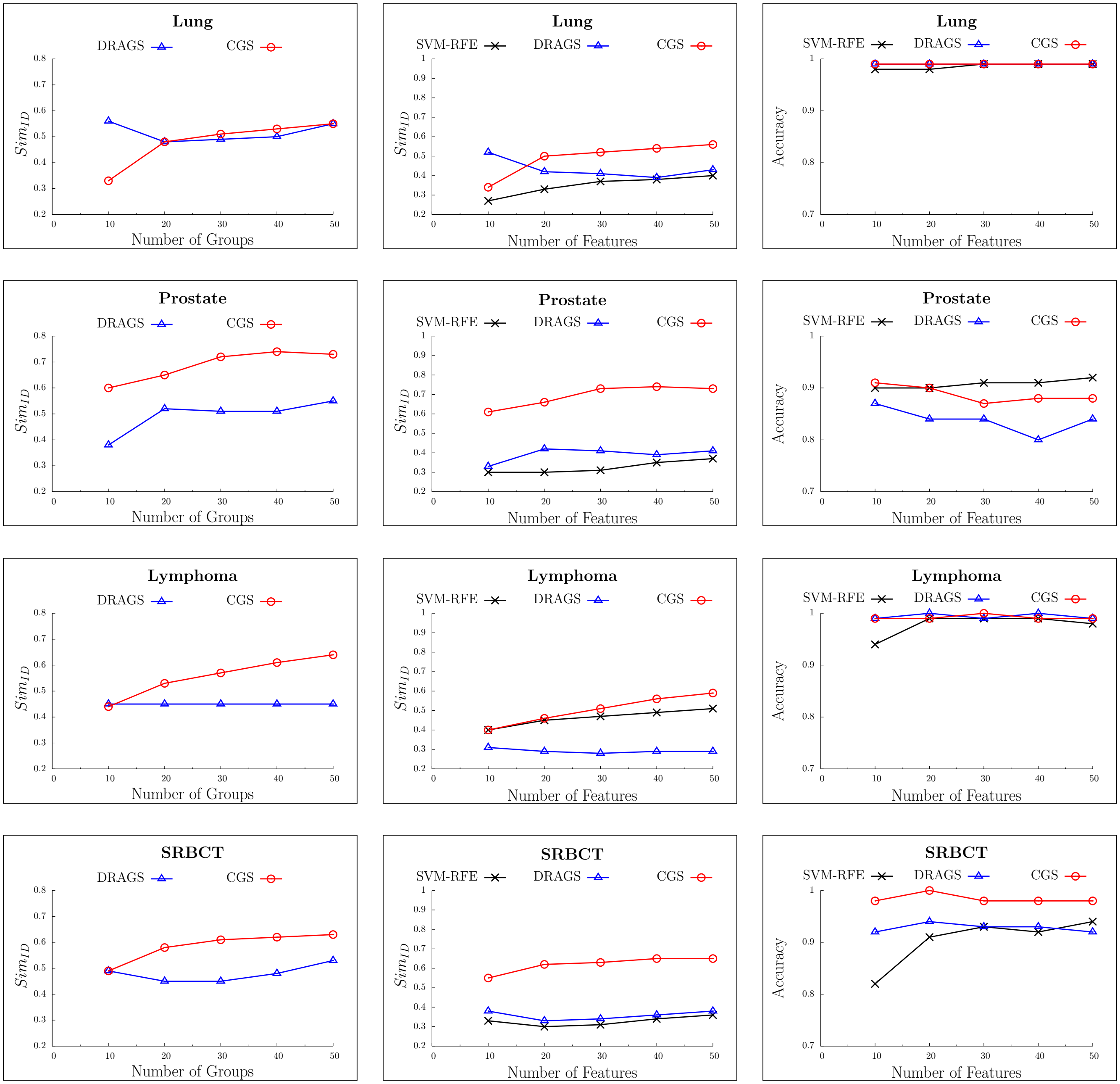


Figure 5: Comparison of SVM-RFE, DRAGS, and CGS on *Lung*, *Prostate*, *Lymphoma* and *SRBCT* microarray data sets. Figures in the left, middle, and right columns respectively show the stability w.r.t. Sim_{ID} of the selected feature groups, the stability w.r.t. Sim_{ID} of the selected representative features, and the SVM classification accuracy, for various numbers of selected features (or groups).