New Gene Subset Selection Approaches Based on Linear Separating Genes and Gene-Pairs

Amirali Jafarian, Alioune Ngom, and Luis Rueda

School of Computer Science, University of Windsor, Windsor, Ontario, Canada
{jafaria,angom,lrueda}@uwindsor.ca

Abstract. The concept of linear separability of gene expression data sets with respect to two classes has been recently studied in the literature. The problem is to efficiently find all pairs of genes which induce a linear separation of the data. It has been suggested that an underlying molecular mechanism relates together the two genes of a separating pair to the phenotype under study, such as a specific cancer. In this paper we study the *Containment Angle* (CA) defined on the unit circle for a linearly separating gene-pair (LS-pair) as an alternative to the paired *t*-test ranking function for gene selection. Using the CA we also show empirically that a given classifier's error is related to the degree of linear separability of a given data set. Finally we propose gene subset selection methods based on the CA ranking function for LS-pairs and a ranking function for linearly separation genes (LS-genes), and which select only among LS-genes and LS-pairs. Our methods give better results in terms of subset sizes and classification accuracy when compared to a well-performing method, on many data sets.

Keywords: Linearly Separating Features, Gene Expression, Microarray, Gene Selection, Feature Ranking, Filtering, Subset Selection.

1 Introduction

DNA microarrays give the expression levels for thousands of genes in parallel either for a single tissue sample, condition, or time point. Microarray data sets are usually noisy with a low sample size given the large number of measured genes. Such data sets present many difficult challenges for sample classification algorithms: too many genes are noisy, irrelevant or redundant for the learning problem at hand. Our present work introduces gene subset selection methods based on the concept of *linear separability* of gene expression data sets as introduced recently in [1]. We use their geometric notion of *linear separation* by pairs of genes (where samples belong to one of two distinct classes termed *red* and *blue* samples in [1]) to define a simple criterion for selecting (best subsets of) genes for the purpose of sample classification. Gene subset selection methods have received considerable attention in recent years as better dimensionality reduction methods than feature extraction methods which yield features that are difficult to interpret. The gene subset selection problem is to find a smallest subset of genes, whose expression values allow sample classification with the highest possible accuracy. Many approaches have been proposed in the literature to solve this problem. A simple and common method is the *filter approach* which first

M. Loog et al. (Eds.): PRIB 2011, LNBI 7036, pp. 50–62, 2011.

ranks single genes according to how well they each separate the classes (we assume two classes in this paper), and then selects the top *r* ranked genes as the gene subset to be used; where *r* is the smallest integer, which yields the best classification accuracy when using the subset. Many gene ranking criteria are proposed based on different (or a combination of) principles, including *redundancy* and *relevancy* [2], [5]. Filter methods are simple and fast, but they do not necessarily produce the best gene subsets; since there are gene subsets allowing better separation than the best subsets of top ranked genes. Other methods introduced in literature are the *wrapper approaches*, which evaluate subsets of genes irrespective of any possible ranking over the genes. Such methods are based on heuristics which directly search the space of gene subsets and guided by a classifier's performance on the selected gene subsets [8]. The best methods combine both gene ranking and wrapper approaches but they are computationally intensive.

Recently, some authors have considered pairs of genes as features to be used in filtering methods rather using than single genes. The motivation for using gene-pairs instead of single genes is that two single genes considered together may distinguish the classes much better than when they are considered individually; this is true even if one or both of the genes have low ranks from a ranking function defined for single genes. In other words, when we select only top-ranked single genes using such ranking function, some subsets of genes which have greater class distinguishing capability (than the subset of top-ranked genes) will not be selected due to the presence of low-ranked single genes. The authors of [2] devised the first gene selection method based on using pairs of genes as features. Given a gene-pair, they used *diagonal linear discriminant* (DLD) and compute the projected coordinate of each sample data on the DLD axis using only the two genes, and then take the two-sample *t*-statistic on these projected samples as the pair's score. The authors then devised two filter methods for gene subset selection based on the pair *t*-scores. Our approach in [10] was to use and evaluate linearly separating pairs of genes (LS-pairs) for the purpose of finding the best gene subsets. We proposed a simple ranking criterion for only LS-pairs and in order to evaluate how well each pair separates the classes. Additionally in order to find the best gene subsets, we devised a filter method, based on selecting only LS-pairs.

Our approach in this paper is to use both linearly separating singles genes (LS-genes) and linearly separating gene-pairs (LS-pairs) as features for the purpose of finding the best gene subsets. We propose ranking criteria for both LS-genes and LS-pairs in order to evaluate how well such features separate the classes then devise methods that select among top-ranked LS-genes and LS-pairs.

2 Linear Separability of Gene Expression Datasets

Recently, [1] proposed a geometric notion of *linear separation* by gene pairs, in the context of gene expression data sets, in which samples belong to one of two distinct classes, termed *red* and *blue* classes. The authors then introduced a novel highly efficient algorithm for finding all gene-pairs that induce a linear separation of the two-class samples. Let $m = m_1 + m_2$ be the number of samples, out of which m_1 are red and m_2 are blue. A gene-pair $g_{ij} = (g_i, g_j)$ is a *linearly separating* pair (LS-pair) if there exists a separating line L in the two-dimensional (2D) plane produced by the projection of the m samples according to the pair g_{ij}; that is, such that all the m_1 red samples are in one side of L and the remaining m_2 blue samples are in the other side of L, and no sample lies on L itself. Figure 1 and 2 show examples of LS and non-LS gene pairs, respectively.

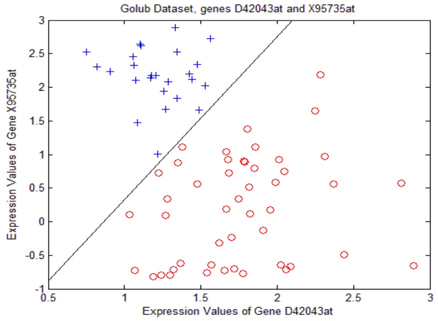

Fig. 1. An LS-pair taken from Golub (Leukemia) dataset

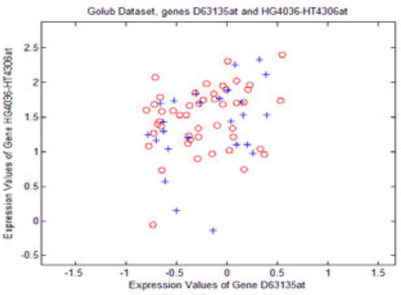

Fig. 2. A non LS-pair taken from Golub (Leukemia) dataset

In order to formulate a condition for linear separability, [1] first views the 2D points in a geometric manner. That is, each point of an arbitrarily chosen class, say red class, is connected by an arrow (directed vector) to every blue point. See Figures 3a and 4a, for example. Then the resulting m_1m_2 vectors are projected onto the unit circle, as in Figures 3b and 4b, retaining their directions but not their lengths. The authors then proceed with a theorem proving that: *a gene pair* $g_{ij} = (g_i, g_j)$ *is an LS pair* if and only if *its associated unit circle has a sector of angle* $\beta < 180°$ *which contains all the* m_1m_2 *vectors*. Figures 3 and 4 illustrate this theorem for pairs (x, y). Thus, to test for linear separability of pair g_{ij} one only needs to find the vector with the smallest angle and the vector with the largest angle and check whether the two vectors form a sector of angle $\beta < 180°$ containing all m_1m_2 vectors.

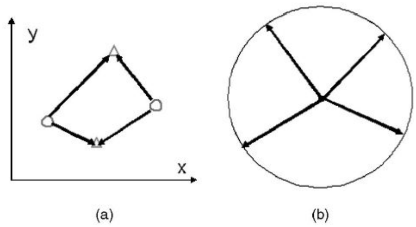

Fig. 3. A set of four non-separable points. (a) The construction of the vectors. (b) Their projection onto the unit circle [1].

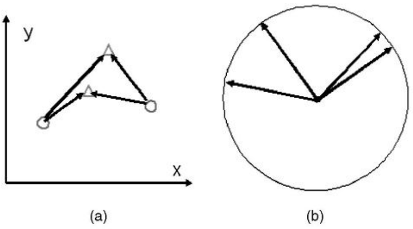

Fig. 4. A set of four separable points producing vectors on the unit circle that are contained in a sector of angle $\beta < 180°$ [1]

Using the theorem above, [1] proposed a very efficient algorithm for finding all LS-pairs of a data set. Next, they derived a theoretical upper bound on the *expected number* of LS-pairs in a *randomly labeled* data set. They also derived, for a given data set, an empirical upper bound resulting from shuffling the labels of the data at random. The degree to which an actual gene expression is linearly separable, (in term of the actual number of LS-pairs in the data) is then derived by comparing with the theoretical and empirical upper bounds. Seven out of the ten data sets they have examined were highly separable and very few were not (see Table 4).

Let G be the set of genes, we generalize the definition of linear separation to apply to any t-tuple $g_{1\ldots t} = (g_{i1}, g_{i2}, \ldots, g_{it})$ of genes where $1 \leq t \leq |G|$, $1 \leq j \leq t$, and $i_j \in \{1, \ldots, |G|\}$, and say that: $g_{1\ldots t}$ is a linearly separating t-tuple (LS-tuple) if there exists a separating (t-1)-dimensional hyperplane H in the t-dimensional sub-space defined by the genes in $g_{1\ldots t}$. It remains open to generalize the theorem of [1] to t-tuples of genes, $t \geq 1$, by considering projecting the m_1m_2 vectors obtained from the t-dimensional points onto a unit (t-1)-sphere, and then determine a test for linearly separability of a t–tuple from the (t-1)-sphere. Clearly, the theorem is true for t=1: since a 0-sphere is a pair of points delimiting a line segment of length 2, and that the m_1m_2 vectors point in the same direction (i.e., they form a sector of angle 0) if and only the single gene is linearly separable.

3 Feature Ranking Criteria

As said before, we will use LS-genes and LS-pairs as features to select from, and for the purpose of finding a minimal number of such features such that their combined expression levels allow a given classifier to separate the two classes as much as possible. Our approach in this paper is to first obtain all the LS-genes and LS-pairs of a given data set, rank these features according to some ranking criteria, and then apply a filtering algorithm in order to determine the best subsets of genes.

3.1 LS-Pair Ranking Criterion

The LS-pairs from given data sets were also used as classifiers in [1], using a standard training-and-test process with cross-validation. The authors compared the performance of these new classifiers with that of an SVM classifier applied to the original data sets without gene selection step. They found that highly separable data sets exhibit low SVM classification errors, while low to non-separable data sets exhibit high SVM classification errors. However, no theoretical proof exists showing the relation between SVM performance and the degree of separability of a data set; although this seems quite intuitive.

In [10], we investigated the relationship between the performance of a classifier applied to an LS-pair of a given data set and the β-sector of the LS-pair (discussed in Section 2, see Fig. 4b). We call β, the *Containment Angle*. Intuitively, the smaller is β for an LS-pair then the higher should be the accuracy of a classifier using the LS pair as input. This is because: the smaller is the angle β, the farther the samples are from the separating line L. Also for LS-pairs, the generalization ability of a classifier should decreases when β is close to 180° since some samples are very close to the separating line. To test this, we used the algorithm of [1] in [10] to generate all the LS pairs of a given data set and sorted them in increasing order of their angles β. We then proceeded as follows. For each LS pair, $g_{ij} = (g_i, g_j)$ of D, we applied a classifier with 10 runs of 10-fold cross-validation on D but using g_{ij} as the feature subset. We experimented on each separable data set examined in [1] and tried with many classifiers. From these experiments, we observed that the accuracy of the classifiers increased in general as the containment angle decreased from the bottom LS-pair (having largest angle) to the top LS-pair (having smallest angle). There were very few examples (see last row of Table 1, for instance), where the accuracy does not increase monotonously as the angle decreases within a consecutive sequence of LS-pairs. However, the average of the accuracies of the bottom ten LS-pairs were lower than that of the top ten LS-pairs. These experiments also show that using LS pairs is a better alternative than using the full set of genes for sample classification purpose, since classifying using pairs is much faster than using the gene set while still giving satisfactory performances. This enforces our intuition above while suggesting that one can use the Containment Angle as a measure of the quality of an LS-pair.

Table 1 shows the performance of SVM used on each of the top three LS-pairs for each data set, and compares with SVM used on all genes of the data sets (last column) with ten-fold cross-validation. In Table 1, we can see that applying SVM on top LS-pairs yields performance comparable to applying SVM on the full gene set; indeed better accuracies are obtained from the LS-pairs than from the full data. (in bold fonts).

3.2 LS-Gene Ranking Criterion

As mentioned earlier, a single gene is an LS-gene if and only if all the m_1m_2 vectors in the corresponding zero-sphere point in the same direction (See Fig. 5 and 6 for a non LS-gene, an LS-gene and their projection in the Zero-sphere). We use a simple ranking criterion illustrated in Fig. 7: for each LS-gene, we compute the quantities A and B and use the ratio A/B as the score of the LS-gene.

4 Gene Subset Selection

Gene subset selection approaches based on gene pairs have been proposed in [2]. For a given gene pair, the authors used a two-sample t-statistic on projected data samples as the score of pairs (pair t-score), and then pairs are ranked according to their t-scores for the purpose of subset selection. They devised two subset selection algorithms which differ in the way gene pairs are selected for inclusion in a current subset. In their fastest method, they iteratively select the top-ranked gene g_i from the current list of genes, then find a gene g_j such that the t-score of the pair $g_{ij} = (g_i, g_j)$ is the maximum given all pairs $g_{ik} = (g_i, g_k)$, and then remove any other gene-pairs containing either g_i or g_j; this continues until r gene are selected. In their best but very slow method, they generate and rank all the possible gene pairs, and then select the top r ranked gene-pairs. The gene-pairs in [2] are not necessarily LS-pairs. In [10], we iteratively selected the top-ranked LS-pair until r genes are selected.

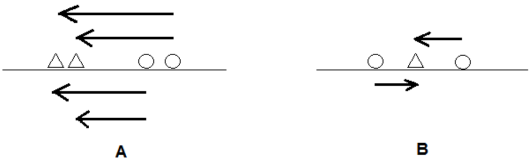

Fig. 5. A set of points causing Linear Separability (Left Panel) Vs. Non Linear Separability (Right Panel)

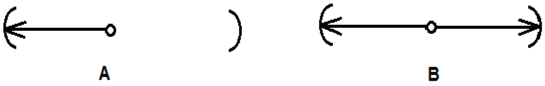

Fig. 6. The projection of vectors of LS Points in the Zero-Sphere (Left Panel) Vs. Non Linear Separability (Right Panel)

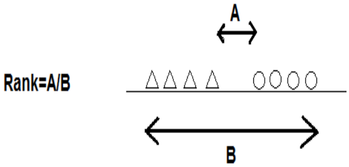

Fig. 7. Ranking Criterion for LS Genes

In this section, we propose gene subset selection approaches based on selecting only LS-genes and LS-pairs. The problem with this is that, initially, a data set may have a low degree of linear separability, and hence, not enough LS-features to select from. To overcome this problem, we first apply SVM with soft margin on the initial given data set before performing any gene selection method, and then sort the support vector (SV) samples in decreasing order of their Lagrange coefficients. When there are no more LS-features to select from during the process of gene selection, we then iteratively remove the current SV sample having the largest Lagrange coefficient, until the resulting data set contains LS-features; such SV samples are farthest from the separating maximum margin hyperplane and are probably misclassified by SVM. We devised two filtering methods to be discussed below.

Our first gene subset selection method (LSGP) proceeds by iteratively selecting in this order, from the LS-genes and then from the LS-pairs until a subset S of r genes is obtained. The LS-features are ranked according to the ranking criteria discussed above. In the LSGP algorithm, S is the subset to be found and r is the desired size of S, and G and P are respectively the sets of LS-Genes and LS-pairs. In lines 6.b and 7.c.ii, we apply a classifier to the currently selected subset S to keep track of the best subset *Best-S* of size $\leq r$. We use ten runs of ten-fold cross-validation on S, and the algorithm returns subsets S and *Best-S* and their performances. SV samples with largest Lagrange coefficients are iteratively removed from data set *D*, in line 8.a.i, whenever there are not enough LS-pairs in the current *D*. When an LS-gene g_i (resp., LS-pair g_{ab}) is selected, we also remove all LS-pairs containing g_i, (resp., g_a or g_b); see lines 7.b and 7.c.iii. This deletion is in order to minimize redundancy. That is, when LS-gene g_i is selected then any LS-pair containing g_i will be redundant. In [2] they select the gene top-ranked gene g_i and then find a gene g_j such that the pair g_{ij} has maximal pair *t*-score. Also in their slow approach (which yields better performance than their fast method) they iteratively select the top-ranked pairs in such a way that the selected pairs are mutually disjoint from each other. That is, they delete all of those pairs which intersect the currently selected subset of genes. This deletion is not true for selected LS-pairs, however. Assume an LS-pair $g_{ab} = (g_a, g_b)$ is selected and assume LS-pair $g_{bc} = (g_b, g_c) \in P$ not yet selected. If we remove g_{bc}, then the *possible* LS-triplet $g_{abc} = (g_a, g_b, g_c)$, which may yield a better subset *S* or a shorter subset *Best-S*, will be lost. Hence, in our second method (DF-LSGP) we consider the intersection graph $N = (P, E)$ where, the vertex set is the set of LS-pairs, *P*, in *D* and edges $(v_i, v_j) \in E$ if v_i and v_j have a gene in common. We then perform a graph traversal algorithm on *N*, which selects LS-pairs as the graph is being traversed.

Table 1. Accuracy on the top three LS-pairs versus accuracy on the full gene set, using SVM with hard margin

	TP1	TP2	TP3	Full Data
Small Beer	98.96%	98.96%	98.96%	**100%**
Beer	98.96%	98.96%	98.96%	**99.06%**
Squamous	**100%**	**100%**	**100%**	**100%**
Bhttacharjee	99.23%	**100%**	99.74%	98.08%
Gordon	99.83%	99.56%	**99.94%**	99.28%
Golub1	95.42%	**100%**	**100%**	98.61%

The differences between our two methods are in lines 7 up to but not including lines 8. In DF-LSGP, the LS-genes are selected first as in the first method. Then we iteratively select the best LS-pair vertex and its un-selected neighbors in a depth-first manner; see line 7.6 and thereafter. This continues until the desired number of genes, *r*, is obtained. We have also implemented a breadth-first traversal of the graph, BF-LSGP, where the neighbors of a selected LS-pair are sent to a queue starting from the top-ranked ones. In practice, we do not create an intersection graph *N* (line 7.4) given that *P* may be very large for some data sets; we simply push or enqueue the top-ranked LS-pair from the initial *P* onto the stack or queue (line 7.6.3) then simulate the graph-traversal algorithm.

LSGP - LS-Gene and LS-Pair Selection on *D*:

```
1.  $S \leftarrow \{\}$
2.  $r \leftarrow$ desired number of genes to select
3.  $d \leftarrow 0$
4.  $G \leftarrow$ set of LS-genes of $D$
5.  $G \leftarrow G - \{g_i$ s.t. $g_i \in S\}$; '–' = set-difference
6.  Repeat
        a.  $S \leftarrow S + \{g_i \leftarrow$ top-ranked LS-gene in $G\}$
                ; '+' = union
        b.  Apply a classifier on $S$ and update Best-$S$
        c.  $G \leftarrow G - \{g_i\}$
        d.  $d \leftarrow d + 1$
    Until $d = r$ or $G = \{\}$
7.  If $d < r$ Then
        a.  $P \leftarrow$ set of LS-pairs of $D$
        b.  $P \leftarrow P - \{g_{ij}$ s.t. $g_i \in S$ or $g_j \in S\}$
        c.  Repeat
                i.   $S \leftarrow S + \{g_{ij} \leftarrow$ top-ranked LS-pair in $P\}$
                ii.  Apply a classifier on $S$ and update Best-$S$
                iii. $P \leftarrow P - \{g_{ij}$ s.t. $g_i \in S$ or $g_j \in S\}$
                iv.  $d \leftarrow d + 2$
            Until $d \geq r$ or $P = \{\}$
8.  If $d < r$ Then
        a.  Repeat
                i.  $D \leftarrow D$ – {SV sample with largest Lagrange coefficient}
            Until $D$ contains LS-features
        b.  Repeat from 4 with the resulting $D$
9.  Return $S$, Best-$S$, and their performances
```

DF-LSGP - Graph-Based LSGP Selection on *D*:

```
1-  $S \leftarrow \{\}$
2-  $r \leftarrow$ desired number of genes to select
3-  $d \leftarrow 0$
4-  $G \leftarrow$ set of LS-genes of $D$
5-  $G \leftarrow G - \{g_i$ s.t. $g_i \in S\}$; '– ' = set-difference
                ; remove already selected LS-genes
6-  Repeat
        a.  $S \leftarrow S + \{g_i \leftarrow$ top-ranked LS-gene in $G\}$
                ; '+ ' = union
        b.  Apply a classifier on $S$ and update Best-$S$
        c.  $G \leftarrow G - \{g_i\}$
        d.  $d \leftarrow d + 1$
    Until $d = r$ or $G = \{\}$
7-  If $d < r$ Then
    7.1.  $P \leftarrow$ set of LS-pairs of $D$
    7.2.  $P \leftarrow P - \{g_{ij}$ s.t. $g_i \in G$ or $g_j \in G\}$
              ; remove LS-pairs containing LS-genes
    7.3.  $P \leftarrow P - \{g_{ij}$ s.t. $g_i \in S$ and $g_j \in S\}$
              ; remove already selected LS-pairs
    7.4.  Construct intersection graph $N = (P, E)$
    7.5.  For each vertex $g_{ij}$: set visited $[g_{ij}] \leftarrow$ false
    7.6.  While there are un-visited vertices and $d < r$ Do:
        7.6.1.  $Stack \leftarrow \{\}$
        7.6.2.  $g_{ij} \leftarrow$ top-ranked vertex in $N$
        7.6.3.  Push $g_{ij}$ onto Stack
        7.6.4.  While $Stack \neq \{\}$ and $d < r$ Do:
            7.6.4.1.  Pop $g_{ij}$ from Stack
            7.6.4.2.  If $g_{ij}$ is un-visited Then
                7.6.4.2.1.  visited$[g_{ij}] \leftarrow$ true
                7.6.4.2.2.  $d \leftarrow |S + \{g_{ij}\}|$
                7.6.4.2.3.  $S \leftarrow S + \{g_{ij}\}$
                7.6.4.2.4.  If $S$ has changed Then
                    7.6.4.2.4.1.  Apply a classifier on $S$ and update Best-$S$
                7.6.4.2.5.  $P \leftarrow P - \{g_{ab}$ s.t. $g_a \in S$ and $g_b \in S\}$
                                ; delete already selected vertices from N
                7.6.4.2.6.  Push all un-visited neighbors of $g_{ij}$ onto Stack starting from the least-ranked ones.
8-  If $d < r$ Then
        a.  Repeat
                i.  $D \leftarrow D$ – {SV sample with largest Lagrange coefficient}
            Until the resulting $D$ contains LS-features
        b.  Repeat from 4 with the resulting $D$
9-  Return $S$, Best-$S$, and their performances
```

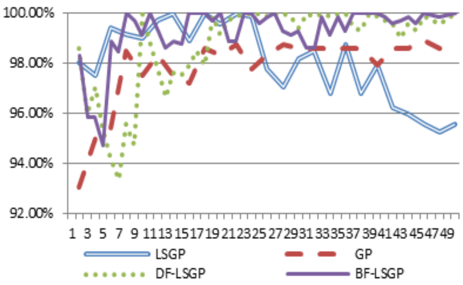

Fig. 8. Performance of SVM-Hard on Golub2

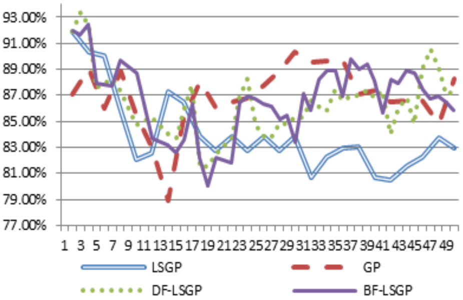

Fig. 12. Performance of SVM-Hard on Alon2

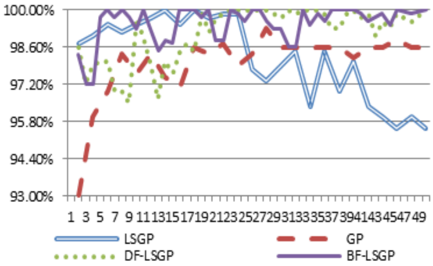

Fig. 9. Performance of SVM-Soft on Golub2

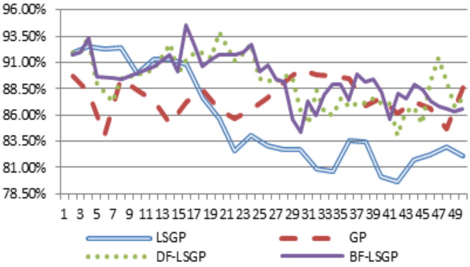

Fig. 13. Performance of SVM-Soft on Alon2

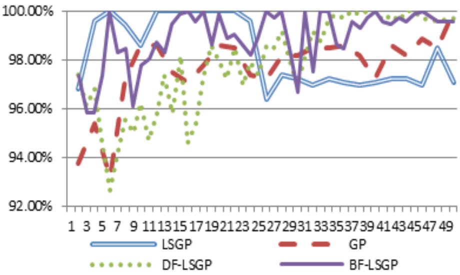

Fig. 10. Performance of KNN on Golub2

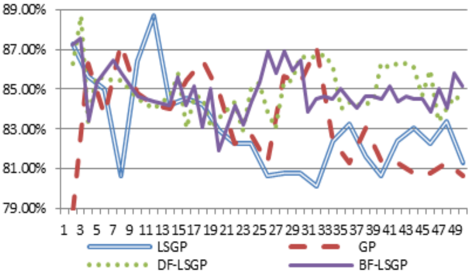

Fig. 14. Performance of KNN on Alon2

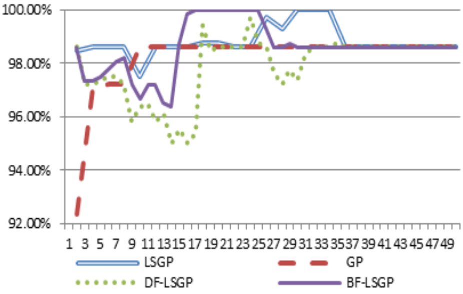

Fig. 11. Performance of DLD on Golub2

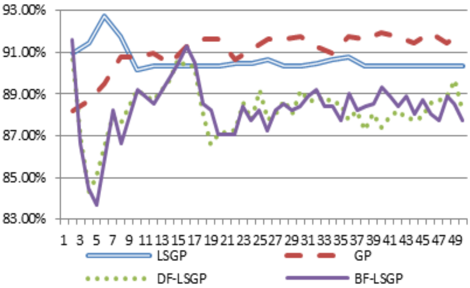

Fig. 15. Performance of DLD on Alon2

5 Computational Experiments

In the first set of experiments, we compared our three filtering approaches (LSGP, DF-LSGP, and BF-LSGP) with the *greedy-pair* (GP) method of [2]. We compared on the two publicly available data sets (Golub [3] and Alon [4]) used in [2]; which we have pre-processed in the same manner as in [2], and renamed as Alon2 and Golub2 to differentiate them with the Golub and Alon data sets used in [1] but pre-processed differently. In these experiments, we set the number of desired genes to $r = |S| \approx 50$ and also keep track of the best subset, *Best-S*, of size $\leq r$. Figures 8 to 15 show the results of our three filtering methods compared with the *greedy-pair* method of [2]. In this set of experiment four classifiers were applied using ten runs of ten-fold cross-validation, and we returned the average accuracy over the hundred folds for both the r-subset S and the best subset *Best-S*. The horizontal axis corresponds to the size of a selected gene subset and the vertical axis is the performance (classifier's accuracy) of the subset. Naturally, the four filtering methods performed best on the *highly separable* Golub2 data set (Figures 8 to 11) and performed worst on the *borderline separable* Alon2 data set (Figures 12 to 15). Our graph-based method, DF-LSGP and BF-LSGP performed better than LSGP and GP, in general; their curves are higher on average. LSGP performed the worst on average. The best subsets *Best-S* returned by our three methods are also smaller than those return by GP. Our graph-based methods make use and take advantage of the information or knowledge already present in the currently selected S subset in order to decide which LS-pairs to select next. Top-ranked LS-pairs which intersect S are always selected first, the advantage of which being the selection of t-tuples which are possibly linearly separating or which give better performances than arbitrarily selected LS-pairs. The selection of LS-pairs in GP and LSGP is somewhat arbitrary since it is based solely on their ranks.

Also we performed the second set of experiments in which the ranking and subset selection are performed on the training dataset within the framework of ten-fold cross-validation process. That is, we partition a data set D into ten distinct parts, and in each iteration of ten-fold cross validation process: 1) we perform feature ranking and selection on the nine-part training set; 2) train a classifier on this training set but using only the selected genes; and 3) estimate the performance of classification on the remaining one-part validation set. We did this set of experiments with our LSGP methods on the eight data sets of [1] (given in Table 4) and which are pre-processed as in [1] also.

The results for these experiments are shown in Tables 2 and 3 for subsets *Best-S* and S respectively. We show the performances of LSGP, DF-LSGP and BF-LSGP in terms of the average accuracy for both subsets S and *Best-S*. We must note that since feature ranking and selection is performed in each fold of the ten-fold cross-validation, then ten different subsets S and *Best-S* are obtained after the ten iterations of the cross-validation process. These subsets are not fixed as in our three sets of experiment above. Thus for subsets *Best-S*, in Table 2, we list in parenthesis the minimum, the average, and the maximum size of the hundred subsets *Best-S* obtained after the ten runs of ten-fold cross-validation, beside showing the average of the accuracies of the hundred subsets. For subsets S, an entry is the average of the accuracies of the hundred subsets of size $r = 50$ each. The averages in both Tables 2 and 3 are quite high, even for the least separable data sets Alon and Adeno Beer. In addition, for all data sets, we obtained a subset *Best-S* with the maximal accuracy of 100%.

6 Benchmark Datasets

To evaluate the performance of our proposed method, we have done extensive experiments on eight publicly available microarray gene expression datasets, namely, Golub [3], Alon [4], Gordon[9], Beer [6], Small Beer [6], AdenoBeer [6], Bhattacharjee [7] and Squamous [7] datasets shown in table 4.

For this research we used eight Datasets which are publicly available. For datasets we did the following preprocessing steps; similar to those dataset used in [1]):

- Trimming: All values lower than 100 were set to 100, and all values higher than 16,000 were set to 16,000, creating a range of 100-16,000.
- Logarithmic transformation: The natural logarithm $\ln(x)$ was taken for each value.
- Standardizing: Each sample was standardized to have a mean of 0 and a standard deviation of 1.

For two other datasets called Golub2 and Alon2 we did the same preprocessing steps, done in [2], in order to have a sound comparison between our Gene Subset returned by our approach and theirs. The preprocessing for these two datasets is as follows:

- Logarithmic transformation: Base 10 logarithmic transformation
- Standardizing: For each gene, subtract the mean and divide by standard deviation.

For Golub2 the following additional preprocessing step is done (Similar to [2]): thresholding with a floor of 1 and filtering by excluding genes with $max\ \ min \leq 500$. This leaves us with a dataset of 3,934 genes.

Due to limited space for the details of all of the datasets used in this research see [1].

Table 2. Accuracy of *Best-S* from [XX]-LSGP, with Ranking and Selection on Training Sets

	KNN			SVM-Soft			SVM-Hard		
	LSGP	**DF-LSGP**	**BF-LSGP**	**LSGP**	**DF-LSGP**	**BF-LSGP**	**LSGP**	**DF-LSGP**	**BF-LSGP**
Beer	100% (1,2.31,36)	99.80% (1,2.46,34)	99.78% (1,2.06,21)	100% (1,2.36,13)	99.60% (1,1.81,11)	99.90% (1,2.16,18)	100% (1,2.16,18)	99.90% (1,2.66,34)	100% (1,2.42,21)
Small Beer	99.18% (1,1.21,3)	98.96% (1,1.18,3)	98.96% (1,1.21,3)	99.18% (1,1.81,3)	98.96% (1,1.13,2)	98.96% (1,1.15,3)	99.69% (1,3.77,32)	99.27% (1,1.90,47)	99.07% (1,1.66,46)
Squamous	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)	100% (1,1,1)
Gordon	99.61% (2,3.76,28)	99.70% (2,4.58,43)	99.77% (2,4.60,47)	99.56% (2,4.12,30)	99.32% (2,4.15,44)	99.52% (2,4.55,40)	99.50% (2,3.88,44)	99.32% (2,3.95,37)	99.84% (2,4.60,40)
Bhttacharjee	98.81% (1,3.41,14)	98.36% (1,2.97,31)	98.19% (1,3.15,44)	98.68% (1,2.68,16)	98.29% (1,2.43,32)	98.11% (1,2.48,44)	98.61% (1,3.06,18)	98.29% (1,2.79,26)	98.18% (1,3.10,46)
Golub	98.01% (2,5.41,42)	98.46% (2,7.79,43)	97.65% (2,6.14,45)	97.70% (2,4.65,48)	97.40% (2,4.96,40)	97.61% (2,5.83,45)	98.67% (2,5.35,30)	99.11% (2,6.1,49)	98.86% (2,6.71,48)
Alon	93.43% (2,7.1,50)	93.93% (2,7.67,45)	94.43% (2,8.26,47)	91.62% (2,6.82,48)	93.57% (2,5.37,36)	92.83% (2,6.24,44)	92.57% (2,6.98,48)	95.19% (2,5.49,35)	94.86% (2,6.99,47)
Adeno Beer	88.33% (2,12.12,50)	88.39% (2,13.53,50)	86.96% (1,10.62,50)	87.64% (1,10.64,48)	87.83% (2,12.52,49)	87.29% (2,12.85,48)	88.38% (2,13.64,48)	88.62% (2,16.07,47)	88.52% (2,14.72,48)

Table 3. Accuracy of *S* from [XX]-LSGP, with Ranking and Selection on Training Sets

	KNN			SVM-Soft			SVM-Hard		
	LSGP	DF-LSGP	BF-LSGP	LSGP	DF-LSGP	BF-LSGP	LSGP	DF-LSGP	BF-LSGP
Beer	99.26%	99.07%	98.94%	99.47%	99.18%	98.94%	99.69%	99.28%	100%
Small Beer	98.98%	98.96%	98.96%	98.98%	98.96%	98.96%	99.08%	99.27%	98.96%
Squamous	100%	100%	100%	100%	100%	100%	100%	100%	100%
Gordon	99.06%	99.09%	99.14%	99.23%	99.02%	98.89%	98.78%	98.72%	98.90%
Bhttacharjee	98.29%	97.33%	96.18%	98.29%	97.65%	96.96%	97.82%	97.63%	97.09%
Golub	95.89%	95.32%	95.31%	96.11%	93.92%	95.23%	96.84%	96.26%	95.99%
Alon	84.57%	85.95%	81.95%	80.57%	80.17%	79.95%	78.45%	80.55%	82.86%
Adeno Beer	75.04%	76.80%	74.83%	74.23%	75.47%	76.29%	73.84%	76.91%	76.17%

7 Conclusion

In this research we investigated the idea of using the concept of linear separability of gene expression dataset for the purpose of gene subset selection. We showed that the Containment Angle (CA) can be used to rank linearly separating pair of genes. We also introduced a new ranking criterion for ranking LS-genes. We proposed three different gene subset selection methods, LSGP, DF-LSGP and BF-LSGP, which select linearly separating features using our ranking criteria. Extensive experiments are carried out showing that our approaches are at least comparable to current filtering methods which are based selecting gene-pairs rather than only single genes.

Table 4. Gene Expression Datasets used

Dataset Name	Cancer Type	Nb of Genes	Nb of Samples	Nb of samples of Class1	Nb of samples of Class2	Degree of Separability
Beer	Lung	7129	96	86	10	High
Small Beer	Lung	4966	96	86	10	Very High
Squamous	Lung	4295	41	21	20	Very High
Gordon	Lug	12533	181	150	31	Very High
Bhttacharjee	Lung	4392	156	139	17	Very High
Golub	Leukemia	7129	72	47	25	High
Alon	Colon	2000	62	40	22	Border Line
Adeno Beer	Lung	4966	86	67	19	No

Our approaches are only *proof of concept* and we are currently studying wrapper methods based on selecting (not necessarily linearly separating) gene-pairs. In this regards, our graph-based methods, DF-LSGP and BF-LSGP, will be modified to back-track or continue the search depending on the classifier's error on the current subset. In this paper we devised ranking criteria applied only to LS-features, which is quite restrictive. Hence, we are devising general ranking criteria which will apply to *all* features, and in such a way that LS-features are ranked very high. As a future research, we plan to generalize the theorem of [1] for generating all linearly separating t-tuples $g_{1\ldots t} = (g_{i1}, g_{i2}, \ldots, g_{it})$ from a given data set and for a given size $t \geq 3$. Finally, we have not cited reported gene subsets obtained by our approaches due to space constraint. In particular for our last set of experiments (Tables 2 and 3) for reporting/returning a *single* gene subset (*S* or *Best-S*) out of the hundred such subsets we obtain after the ten runs of ten-fold cross-validation, we can either 1) take the genes that appear most often in all hundred cross-validation folds, or 2) take the subset that is closest to all the other subsets (centroid) using an appropriate distance measure between subsets.

References

1. Unger, G., Chor, B.: Linear Separability of Gene Expression Datasets. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(2) (April-June 2010)
2. Bø, T.H., Jonassen, I.: New Feature Subset Selection Procedures for Classification of Expression Profiles. Genome Biology 3(4), 0017.1–0017.11 (2002)
3. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeeck, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
4. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: 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)
5. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
6. Beer, D.G., et al.: Gene-Expression Profiles Predict Survival of Patients with Lung Adenocarcinoma. Nature Medicine 8(8), 816–824 (2002)
7. Bhattacharjee, A., et al.: Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinoma Subclasses. Proc. Nat'l Academy of Sciences of the USA 98(24), 13790–13795 (2001)
8. Kohavi, R., John, G.: Wrapper for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
9. Gordon, G.J., et al.: Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. Cancer Research 62(17), 4963–4967 (2002)
10. Jafarian, A., Ngom, A.: A New Gene Subset Selection Approach Based on Linear Separating Gene Pairs. In: IEEE International Conference on Computational Advances in Bio and medical Sciences (ICCABS 2011), Orlando FL, February 3-5, pp. 105–110 (2011)