Boost Feature Subset Selection: A New Gene Selection Algorithm for Microarray Dataset

Xian Xu and Aidong Zhang

State University of New York at Buffalo, Buffalo, NY 14260, USA
{xianxu, azhang}@cse.buffalo.edu

Abstract. Gene selection is usually the crucial first step in microarray data analysis. One class of typical approaches is to calculate some discriminative scores using data associated with a single gene. Such discriminative scores are then sorted and top ranked genes are selected for further analysis. However, such an approach will result in redundant gene set since it ignores the complex relationships between genes. Recent researches in feature subset selection began to tackle this problem by limiting the correlations of the selected feature set. In this paper, we propose a novel general framework BFSS: Boost Feature Subset Selection to improve the performance of single-gene based discriminative scores using bootstrapping techniques. Features are selected from dynamically adjusted bootstraps of the training dataset. We tested our algorithm on three well-known publicly available microarray data sets in the bioinformatics community. Encouraging results are reported in this paper.

1 Introduction

Recent technological advances in large scale DNA profiling enable researchers to simultaneously monitor the expression levels of thousands of genes or ESTs [1, 6, 10]. This provides unique opportunities to uncouple the relationship between disease phenotypes and their biochemical causations. As a first step, researchers have attempted to classify disease using such molecular evidences [1, 6, 10].

Microarray technology also raises new challenges for data analyzing algorithms because of the uniqueness of the resulting microarray dataset. In a typical microarray study, the genes or ESTs being monitored number from thousands to tens of thousands, while the number of different tissue samples is much smaller ranging from tens to hundreds. This results in the situation where the number of features (or genes) well outnumbers the number of observations. The term "peaking phenomenon" is coined in the machine learning and pattern recognition community, referring to the phenomenon that inclusion of excessive features may actually degrade the performance of a classifier if the number of training examples used to build the classifier is relatively small compared to the number of features [8]. Gene selection is commonly performed before sample classification is even attempted to alleviate the above stated problem. A smaller subset of informative genes is also a good start point of further biological investigations. The broadly used gene selection algorithms on microarray data sets share a common workflow: 1. some single-gene based discriminative score is selected; 2. genes are ranked based on such discriminative score; and 3. top scored genes are then selected

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 670–677, 2006.

for further investigation. We term this class of algorithms single-gene based gene selection. Although relatively simple, various single-gene based gene selection algorithms have been proposed and demonstrated to be effective for improving sample classification accuracy. Some of them are statistical tests (t-test, F-test) [3], non-parametric tests like TNoM [2], mutual information [13], S2N ratio (signal to noise ratio) [6] etc.

However, the assumption of independence between genes oversimplifies the complex relationship between genes. Genes are well known to interact with each other through gene regulative networks. As a matter of fact, the common assumption of popular cluster analysis on microarray data sets [9] is that co-regulated genes have similar expression profiles. Several of recent researches on feature subset selection especially gene selection [7, 11, 12, 15] explicitly took into consideration the correlations among features (genes) by limiting redundancy in resulting feature (gene) set. We showed in earlier research that the concept of virtual gene (correlations between genes) [14] could help improve gene selection.

In this work, we propose a novel meta-algorithm for boosting the performance of single-gene based gene selection algorithms. In our framework, genes are selected not from original training samples, but from its bootstraps. The probability table of sampling for the different training samples is dynamically adapted based on how previous selected genes behave. Our proposed framework is a general purpose meta-algorithm. Instead of tying with some fixed single-gene based discriminative scores, our algorithm accepts as input most if not all single-gene based gene selection algorithms and produces a better gene selection algorithm. It is worth mentioning that our algorithm is closely related to bagging [4] and boosting [5] proposed for ensemble classifier design.

The rest of this paper is organized as follows. Our proposed algorithm is discussed in Section 2 in detail with an illustrating example. Extensive experimental results on three publicly available microarray data sets are reported in Section 3. In Section 4, we conclude this paper and give directions of our future work.

2 BFSS: Boost Feature Subset Selection

In this section we formulate our boost feature subset selection algorithm (BFSS). Notation used throughout this paper is introduced in the first subsection. An illustrating example is given in subsection two. Detailed algorithm is given in the third subsection.

2.1 Notation

Let $\mathcal{R}$ be the set of real numbers and $\mathcal{N}$ be the set of natural numbers including 0. Let $\mathcal{G} = \{g_1, g_2, \cdots, g_n\}$ be the set of all genes that are used in one study, $\mathcal{S} = \{s_1, s_2, \cdots, s_m\}$ be the set of all experiments performed, $\mathcal{L} = \{l_1, l_2, \cdots, l_l\}$ be the set of sample class labels of interest. We assume $\mathcal{G}, \mathcal{S}, \mathcal{L}$ are fixed for any given study. Let $n = |\mathcal{G}|$ be the total number of genes, $m = |\mathcal{S}|$ be the total number of experiments and $l = |\mathcal{L}|$ be the total number of class labels. A microarray gene expression dataset used in our study can be defined as $\mathcal{E} = (\mathcal{G}, \mathcal{S}, \mathcal{L}, L, E)$, where L is a function $\mathcal{S} \to \mathcal{L}$ such that for $s \in \mathcal{S}$, $L(s) \in \mathcal{L}$ is the class label for sample s; E is a function $\mathcal{G} \times \mathcal{S} \to \mathcal{R}$. For $g \in \mathcal{G}$ and $s \in \mathcal{S}$, $E(g,s)$ is the expression level of gene g in experiment s. In the

bioinformatics community, the function E is normally presented as a two dimensional array of real numbers.

Sometimes we need to treat the set of samples $\mathcal{S}$ as a multiset (or bag). In this case we refer to the set of samples $\mathcal{S}$ as $\mathcal{S}^{\mathcal{M}_1} = (\mathcal{S}, \mathcal{M}_1)$, where $\mathcal{M}_1$ is a function that is always 1 ($\mathcal{M}_1(s) = 1, s \in \mathcal{S}$). A multiset is a set that allows duplication. In the case of $\mathcal{S}^{\mathcal{M}_1}$, $\mathcal{S}$ is the underlying set and $\mathcal{M}_1$ is the multiplicity function for elements in the underlying set. More generally we refer to multiset $\mathcal{S}^{\mathcal{M}} = (\mathcal{S}, \mathcal{M})$ as the **bootstrap sample set**, where $\mathcal{M}$ is an arbitrary function $\mathcal{S} \rightarrow \mathcal{N}$. We will discuss **bootstrap sample set** in detail later in this section.

For simplicity of presentation, we use a subscripting scheme to refer to elements in $\mathcal{E}$. Let $\mathcal{E}(G, \mathcal{S}^{\mathcal{M}}) = (G, \mathcal{S}^{\mathcal{M}}, L, E)$ where $G \subseteq \mathcal{G}$ and $\mathcal{S}^{\mathcal{M}}$ is a **bootstrap sample set**. We further use $L(\mathcal{S}^{\mathcal{M}})$ to denote the set of class labels for the set of experiments $\mathcal{S}^{\mathcal{M}}$.

2.2 Motivating Example

The most obvious drawback of single-gene based gene selection algorithms is the fact that those algorithms ignore the relationships that exist between genes. There is no guarantee that the combination of two "good" features will necessarily produce a "better" classifier. As an illustrating example, please refer to Figure 1, in which the expression levels of three genes across 100 samples are plotted. Samples are labeled using two class labels: either caner (grey background) or normal (white background).

The first two genes, gene 1 and gene 2 behave similarly. In majority of the samples (samples 1 to 40 and samples 61 to 100, or 80% of samples), the expression levels of gene 1 or gene 2 can be used to predict sample class labels effectively. Actually the expression levels of these two genes are generally higher in cancer samples than in normal ones. However, the expression levels of these two genes in samples 41 to 60 (20% of samples) are more mixed across cancer/normal class distinction.

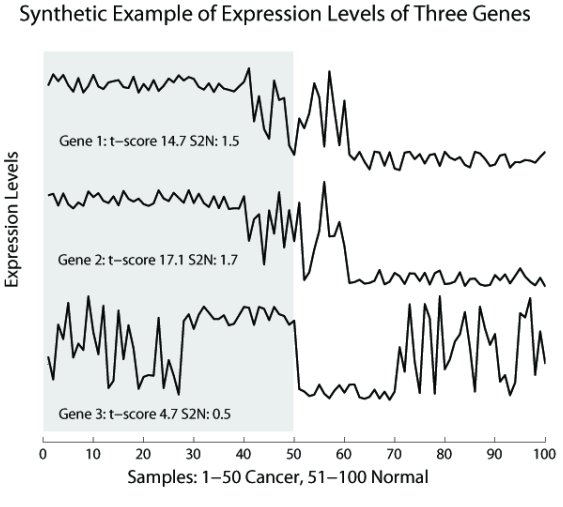

Fig. 1. Illustrating example of redundancy in selected gene set

Gene 1 and gene 2 score high in term of t-score and S2N scores as shown in Figure 1. Gene 3 is obviously a less capable predictor when considered alone. Clear trend exists in the expression levels of gene 3 in samples 31 to sample 70 (40% of samples). However it varies across cancer/normal labels in the rest samples (60%). Gene 3 scores much lower than gene 1 and gene 2 in term of t-score and S2N as expected. Using t-score and S2N, we can rank these three genes based on their salience in predicting cancer/normal class labels as: gene 2 $>$ gene 1 $>$ gene 3.

However, t-score and S2N do not consider the fact that gene 1 and gene 2 behave similarly. They both work well in samples 1 to 40 and samples 61 to 100. They both share more difficult samples, namely samples 41 to 60. If two genes out of the three are to be selected for further data analysis, would it be wise to use both gene 1 and gene 2, as suggested by their relatively high single-gene based discriminative score (t-score and S2N) ranking? This is the very question we address in this paper. We empirically show that it is not the case. Choosing gene 2 and gene 3 might be a better idea as gene 3 "covers" the more difficult samples where gene 2 fails.

2.3 BFSS: Boost Feature Subset Selection

In this section, we elaborate our new algorithm. First we will define some concepts and then describe our BFSS algorithm.

A **bootstrap sample set** $\mathcal{S}^{\mathcal{M}} = (\mathcal{S}, \mathcal{M})$ is a multiset of samples randomly drawn with replacement from the original set of samples $\mathcal{S}$. $\mathcal{M}(s), s \in \mathcal{S}$ is the multiplicity of item s. As a result, the same sample $s \in \mathcal{S}$ can appear more than once or does not appear at all in $\mathcal{S}^{\mathcal{M}}$. The cardinality of $\mathcal{S}^{\mathcal{M}}$ is denoted by m_b. The sampling probability of each sample in $\mathcal{S}$ is determined by a probability table $p(s)$ where $s \in \mathcal{S}$.

Definition 1. A bootstrap sample set *$\mathcal{S}^{\mathcal{M}} = (\mathcal{S}, \mathcal{M})$ of size m_b is a multiset of samples resulting from random sampling from $\mathcal{S}$ with replacement. The probability of each sample $s \in \mathcal{S}$ being sampled is $p(s)$. $m_b = \sum_{s \in \mathcal{S}} \mathcal{M}(s)$.*

Definition 2. ***A bootstrap*** *$\mathcal{B}$ of a training dataset $\mathcal{E} = (\mathcal{G}, \mathcal{S}, L, E)$ using bootstrap sample set $\mathcal{S}^{\mathcal{M}}$ is a dataset defined as*

$$\mathcal{B} = (\mathcal{G}, \mathcal{S}^{\mathcal{M}}, L, E)$$

Definition 3. *The* **worst set of samples** *$\mathcal{S}_{worst}$ of size δ with respect to bootstrap dataset $\mathcal{E}(g, \mathcal{S}^{\mathcal{M}})$ and a single-gene based scoring function F is defined as a multiset:*

$$\underset{S \subseteq \mathcal{S}^{\mathcal{M}} \textbf{ and } |S| = \delta}{argmax} \left(F(\mathcal{E}(g, \mathcal{S}^{\mathcal{M}} - S))\right)$$

Here $\mathcal{S}^{\mathcal{M}} - S$ means a set by removing S from $\mathcal{S}^{\mathcal{M}}$. We also call $\mathcal{S}^{\mathcal{M}} - \mathcal{S}_{worst}$ the **best set of samples**.

A **bootstrap** of training set is defined using the definition of **bootstrap sample set**. A bootstrap $\mathcal{B}$ is a new dataset defined using the four-tuple notation we used to define microarray dataset. The only difference is that the second element is a bootstrap sample

Algorithm 1. *WorstSampleSet* : Calculate the **worst set of samples** using a greedy algorithm

Require: $\mathcal{E} = (\{g\}, S, L, E)$, F as a single-gene based discriminative score
Ensure: S' is the **worst set of samples** with respect to $\mathcal{E}$ and F
1: S', S_0 to be empty sets
2: **for all** $s \in S$ **do**
3: $S_1 \leftarrow S - \{s\}$
4: calculate $F(\mathcal{E}(\{g\}, S_1))$, add score to S_0
5: **end for**
6: sort S_0, add samples s corresponding to top δ scores in S_0 to S'
7: **return** S'

Algorithm 2. *BFSS* : Boost Feature Subset Selection

Require: $\mathcal{E} = (G, S, L, E)$; n' as the number of genes to be selected; F as a single-gene based discriminative score
Ensure: G' as the selected gene set by BFSS using F.
1: Initialize $p(s)$ to be $1/m$ (m is the number of samples in $\mathcal{E}$). Set G' as an empty set.
2: $\mathcal{E}' \leftarrow \mathcal{E}$
3: **for** $|G'| < n'$ **do**
4: generate bootstrap sample set $\mathcal{S}^{\mathcal{M}}$ and bootstrap $\mathcal{B}$ of training set $\mathcal{E}'$ by random sampling using probabilities $p(s)$
5: calculate score F on bootstrap $\mathcal{B}$, refer to this score as F', keep track of the best score so far
6: add top ranked gene g based on F' to G'
7: find worst δ samples $\mathcal{S}_{worst}$ based on $\mathcal{B}(g, \mathcal{S}^{\mathcal{M}})$ using Algorithm 1
8: reduce $(p(s)$ where $s \in \mathcal{S}^{\mathcal{M}} - \mathcal{S}_{worst}$ by a factor of ε and normalize $p(s)$ so that it represents a distribution
9: remove g from $\mathcal{E}'$
10: **end for**
11: **return** G'

set. $\mathcal{B}$ shares the same set of genes $\mathcal{G}$, same sample class label mapping L, and same expression levels mapping E with $\mathcal{E}$.

Given a bootstrap $\mathcal{B}$, a gene g and a score function F, the **worst set of sample** of size δ is a set of samples such that by removing them from the dataset $\mathcal{B}$, best F score for gene g is achieved. We refer to all other samples in $\mathcal{S}^{\mathcal{M}}$ other than those in the **worst set of samples** the **best set of samples**. Both the **worst set of samples** and the **best set of samples** are multisets.

By definition of the **worst set of samples** with respect to gene g and score F, it is exponentially hard to find such set of samples since the power set of samples needs to be examined. We employ a simply greedy algorithm as described in Algorithm 1. For each sample $s \in \mathcal{S}^{\mathcal{M}}$, scores F for gene g on each sample set $\mathcal{S}^{\mathcal{M}} - \{s\}$ are computed. Such scores are ranked and the samples corresponding to the best δ scores are treated as the **worst set of samples**. δ is one parameter of our algorithm. However, as shown later in this section, fixed value of δ is used for all datasets we tested with good results.

Boost feature subset selection algorithm (BFSS) is shown in Algorithm 2. After some initialization, the algorithm first generates a bootstrap $\mathcal{B}$ of training set $\mathcal{E}$, where i iteration counter. This involves the generation of **bootstrap sample set** of size m_b and then the **bootstrap** itself. This involves random sampling with replacement from S using probability table $p(s)$.

After **bootstrap** $\mathcal{B}$ of a training set is generated, the F score is then calculated for each gene in $\mathcal{B}$. Best F score so far is kept during the computation, so is the gene associated with it. In the next step, the gene with best F score for current bootstrap B is selected and added to the selected gene set. Based on the selected gene, BFSS then identifies the **worst set of samples** with respect to the currently selected gene and the single-gene based scoring function F using Algorithm 1. The probability table $p(s)$ for generating bootstraps is modified by reducing the probabilities for the **best set of samples** by a constant factor. The probability of those good samples being selected in subsequent analysis is thus reduced, focusing BFSS onto those samples that previously selected genes would not perform well. The currently selected gene is then marked as selected and not considered further by the BFSS algorithm. BFSS repeats this process until n' genes are selected.

There are three parameters m_b (size of **bootstrap sample set**), δ (size of **worst set of samples**) and ε (the degradation factor of sampling probability) used in our algorithm. We experimentally chose δ to be 0.96 of the number of training samples in a dataset and ε to be 0.96. We set m_b to be twice the number of samples in the training set so that a **bootstrap** is more representative. After fixing these three parameters, there is virtually no more need of tweaking our BFSS algorithm. We used these same parameters for all the three data sets we experimented with and achieved good performance on all of them. This indicates BFSS's good property of requiring little tuning for different data sets. We omit complexity analysis of our algorithm in this paper due to space constraint.

3 Experiments

We performed extensive experiments on three publicly available microarray datasets: Colon Cancer [1], Leukemia [6] and multi-class cancer [10]. Data preprocessing is the same as described in [14]. Performance of classifiers is used as a measure of the performance of feature subset selection algorithms. In order not to be biased on which classifiers we use, three very different general purpose classifiers are used: DLD (diagonal linear discriminant), KNN (k-nearest neighbor, k=3) and SVM (support vector machine, with radial kernel). A 2 fold cross-validation procedure is used to estimate classification performance. We systematically examine the performance of different gene subset selection algorithms using these three classifiers with different number of genes selected. The number of genes selected ranges from 2 to 100 in the increment of 2. All experiments are repeated 100 times to get more accurate results.

This paper focuses on improving performance of single-feature based feature subset selection algorithms. Two simple widely used algorithms are used for testing: t-score and S2N. We plug these two algorithms into our BFSS algorithm and refer to the resulting new algorithms as "boost t-score" and "boost S2N" respectively. We also report classification performance without using gene selection algorithms as a comparison.

Table 1. Performance(%) of single-gene based methods and their boosted versions

Num of genes	*FSS method*	*Colon Cancer*			*Leukemia*			*Multi-class*		
		KNN	*DLD*	*SVM*	*KNN*	*DLD*	*SVM*	*KNN*	*DLD*	*SVM*
N/A	No FSS	75.9	64.7	71.3	93.1	65.3	92.0	85.6	77.2	83.4
20	t-score	81.58	80.10	81.35	92.75	94.97	94.25	78.34	78.13	79.92
	Boost t-score	82.81	81.35	82.87	93.67	94.69	94.47	78.86	78.51	80.47
	S2N	81.23	79.94	81.30	93.33	95.19	94.56	78.56	78.26	80.71
	Boost S2N	82.90	81.71	82.16	94.11	95.11	94.94	78.85	78.37	81.14
50	t-score	81.52	78.10	81.81	94.14	95.47	95.19	81.57	78.65	82.13
	Boost t-score	83.87	81.00	83.71	94.47	95.64	94.94	81.93	78.96	82.62
	S2N	81.84	77.39	82.65	94.31	95.77	95.44	81.73	78.84	82.46
	Boost S2N	83.32	81.55	83.19	94.94	95.86	95.00	81.83	79.14	82.72
100	t-score	82.13	76.65	81.74	95.11	95.81	94.36	83.08	78.97	83.30
	Boost t-score	84.39	80.39	83.65	95.75	96.22	95.22	83.21	79.41	83.73
	S2N	81.52	76.39	81.45	95.42	96.00	95.61	83.29	79.19	83.21
	Boost S2N	83.84	80.19	83.68	95.75	96.17	95.81	83.09	79.50	83.63

Table 1 shows part of our experimental result. Boosted version of single gene based discriminative score yields higher classification accuracy in most cases in our experiments. Significant improvement is observed in colon cancer data set, where performance improved as much as 4%. Less yet consistent performance gain is observed in the other two data sets we tested. For Leukemia data set, the average performance is already in mid 90 percent, where the space for improvement is limited. For multi-class data set we tested, classification performance of KNN classifier is actually better than any of the gene selection algorithms in our experiment. Considering there are more than 16000 genes in this dataset, this seems indicating selecting top 100 genes may not be enough. Multi-class data set is also more heterogeneous since the cancer tissues came from 14 common cancer types. It is reasonable to expect more genes are needed to characterize such vast different phenotypes.

4 Conclusion and Future Work

In this paper, we presented a novel general feature subset selection framework to improve the performance of single-gene based discriminative scores. In our approach, genes are selected from bootstraps of training set instead of training set itself. The sampling probability is dynamically adapted based on the performance of previously selected genes on different bootstrap samples. Extensive experiments were performed on three publicly available microarray datasets. According to our experiments, boosted versions of those single-gene based discriminative scores perform consistently better in most cases and in many cases boosted versions perform considerably better than the original scores. A nice feature of our approach is that most if not all single-gene based discriminative scores can be plugged into our system and the resulted BFSS feature selectors are expected to perform better than the original scores according to our experiments. Our approach is also independent of the classifier used.

Although not targeted as feature selection algorithm for ensemble classifiers, our algorithm may work well with such classifiers nonetheless. Better performance could be achieved by diversifying the feature set selected for each member classifier. Since our BFSS algorithm is based on bootstraps of the training set and sampling probability is dynamically adapted, diversity of the selected features is already built in. It is also interesting to examine the behavior of BFSS in conjunction with these ensemble classifiers, especially these bootstrap based ensemble classifiers (bagging and boosting). Furthermore, since BFSS and bootstrap based ensemble classifiers are all based on the bootstrapping concept, it is possible that they can be combined into a uniform framework. We are currently researching into these directions.

References

1. U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.*, 96(12):6745–50, 1999.
2. A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. volume 7, pages 559–83, 2000.
3. T.H. Bø and I. Jonassen. New feature subset selection procedures for classification of expression profiles. *Genome Biology*, 3(4):research0017.1–0017.11, 2002.
4. Leo Breiman. Bagging predictors. *Machine Learning*, 1996.
5. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *in Proc. ICML 1996*, 1996.
6. T. R. Golub et al. Molecular classifications of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7, 1999.
7. J. Jaeger, R. Sengupta, and W. L. Ruzzo. Improved gene selection for classification of microarrays. In *Proc. PSB*, 2003.
8. Anil K. Jain, Robert P.W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.
9. D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.
10. S. Ramaswamy, P. Tamayo, R. Rifkin, S Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T.R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 98(26):15149–15154, 2001.
11. Yuhang Wang, Fillia S. Makedon, James C. Ford, and Justin Pearlman. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21(8):1530–1537, 2005.
12. Y. Wu and A. Zhang. Feature selection for classifying high-dimensional numerical data. In *IEEE Conference on Computer Vision and Pattern Recognition 2004*, volume 2, pages 251–258, 2004.
13. E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proc. 18th International Conf. on Machine Learning*, pages 601–608. Morgan Kaufmann, San Francisco, CA, 2001.
14. Xian Xu and Aidong Zhang. Virtual gene: Using correlations between genes to select informative genes on microarray datasets. *LNCS Transactions on Computational Systems Biology II, LNBI 3680*, pages 138–152, 2005.
15. L. Yu and H. Liu. Redundancy based feature selection for microarray data. In *Proc. of SIGKDD*, 2004.