BIOINFORMATICS

Vol. 17 Suppl. 1 2001
Pages S306–S315

CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts

Eric P. Xing and Richard M. Karp

Division of Computer Science, University of California, Berkeley, Berkeley, CA 94720, USA

Received on February 5, 2001; revised and accepted on April 1, 2001

ABSTRACT

We present CLIFF, an algorithm for clustering biological samples using gene expression microarray data. This clustering problem is difficult for several reasons, in particular the sparsity of the data, the high dimensionality of the feature (gene) space, and the fact that many features are irrelevant or redundant. Our algorithm iterates between two computational processes, feature filtering and clustering. Given a *reference partition* that approximates the correct clustering of the samples, our feature filtering procedure ranks the features according to their intrinsic discriminability, relevance to the reference partition, and irredundancy to other relevant features, and uses this ranking to select the features to be used in the following round of clustering. Our clustering algorithm, which is based on the concept of a normalized cut, clusters the samples into a new reference partition on the basis of the selected features. On a well-studied problem involving 72 leukemia samples and 7130 genes, we demonstrate that CLIFF outperforms standard clustering approaches that do not consider the feature selection issue, and produces a result that is very close to the original expert labeling of the sample set.

Contact: epxing@cs.berkeley.edu

INTRODUCTION

Cluster analysis of gene expression microarray data is a key step in understanding how the activity of genes varies during biological processes and is affected by disease states and cellular environments. Clustering can be used to group genes according to their expression in a set of samples (Eisen et al., 1998; Wen et al., 1998). Ideally, each of the resulting groups should have a coherent expression pattern, possibly suggesting a modular structure in the gene regulation system. Another type of clustering, which is in a sense more difficult because of the *curse of dimensionality* (due to small sample volume and high feature dimensionality), but is very valuable in clinical as well as mechanistic study, is to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes, such as clinical syndromes or cancer types (Golub et al., 1999). In a typical biological system, it is often not clearly known how many genes are sufficient to fully characterize a macroscopic phenotype. But practically, a working mechanistic hypothesis that is testable and largely captures the biological truth seldom involves more than a few dozens of genes, and knowing the identity of these relevant genes is just as important as finding the grouping of samples they induce. Thus, it is essential to formulate the problem of biological pattern recognition of microarray data in a way that involves an interplay between clustering to produce a sample partition and feature selection to identify genes that significantly contribute to the partition of interest.

There is a rich literature on cluster analysis and various techniques have been developed. Several recent reports have shown the application of some of these techniques to the cluster analysis of gene expression data (see (Szallasi and Somogyi, 2001) for an overview). However, few of these works address the issue of feature selection explicitly, perhaps because it does not appear as a serious problem as long as the number of features is relatively small, and few of the features are irrelevant or redundant. This is often the case when the objects to be clustered are genes and the features used to cluster them correspond to a well selected set of samples. The situation is quite different when the objects to be clustered are samples and the features correspond to genes. The failure to recognize the fundamental asymmetry between these two situations may account for the lack of attention to feature selection. In many typical microarray data sets the *sample space* and the *gene space* are of very different dimensionality ($10^1 \sim 10^2$ samples versus $10^3 \sim 10^4$ genes). Furthermore, the design of sample space and gene space (for different clustering purposes) are subject to different levels of quality control. For example, one usually has a clear knowledge of the biological scenario (*e.g.* a cell cycle) in which one wishes to analyze gene expression, and can construct the sample space accordingly (*e.g.*, by taking

time-course data over a cell cycle); on the other hand, when analyzing a sample set (*e.g.*, a patient group), one usually has little knowledge about how to construct an informative gene space because what genes are relevant is unclear. A frequent alternative is to use a complete list of all known genes. The sparsity of the data, the high dimensionality of the feature space, and the fact that many features are irrelevant or redundant cause the following difficulties:

1. There may be many different well-founded, statistically significant ways to cluster samples. A clustering algorithm is not guaranteed to capture a 'meaningful' partition corresponding to some phenotype(s) of actual empirical interest, such as having or not having a particular type of tumor, because the same set of samples may also display gender, age, or other disease variability, which may also serve as partitioning criteria.
2. Microarrays are not typically task-specific and most of the features are not necessarily related to the phenotype of interest. Thus, even when the phenotype of interest, such as tumor type, induces a strong discriminating pattern in the feature space, the distance calculation between samples is still subject to interference from the large number of irrelevant features.
3. The goal of clustering is often not merely to find out the underlying grouping of samples, but also to form some generalizable cluster representations and sample recognition rules so that future novel samples can be correctly labeled. Vapnik and Chervonenkis (1971) showed that the generalization risk bound of such representations and rules increases with its VC dimension, which is exponentially related to the dimensionality of the feature space. Thus, large feature sets inevitably increase the possibility of predictive error based on clustering results.

Several approaches have been taken to selecting features for microarray sample clustering. One approach is to have domain experts select the features. This is obviously not easily generalizable. Another approach is to use a clustering algorithm to group the features into coherent sets and then project the samples onto a lower-dimensional space spanned by the average expression patterns of the coherent feature sets (Hastie et al., 2000). This approach only deals with the feature redundancy problem, but fails to detect non-discriminating or irrelevant features. Principal component analysis (PCA) may remove non-discriminating and irrelevant features by restricting attention to so-called eigenfeatures corresponding to the large eigenvalues, but each basis element of the new feature subspace is a linear combination of all the original features, making it difficult to identify the important features (Hastie et al., 2000; Alter et al., 2000).

The main difficulty of direct feature selection in cluster analysis is the lack of reference information for feature evaluation. In the machine learning literature, feature selection is primarily applied under the supervised learning paradigm (Kohavi and John, 1997; Langley, 1994). The quality of a feature is usually measured with respect to a *reference partition*. The more relevant a feature is to the reference partition, the better it is. In the clustering paradigm, such reference information is not given, making it hard to tell whether a feature is qualified to be included in the analysis.

In this paper, we propose a novel algorithm, CLIFF (Clustering via Iterative Feature Filtering), which combines a clustering process and a feature selection process in a bootstrap-like iterative way, where each process uses the output of the other as an approximate input, and the outputs of the two processes improve hand-in-hand over the course of the iterations.

CLIFF requires both an efficient clustering algorithm and filters for feature selection. We apply a graph partition algorithm, known as Approximate Normalized Cut (Shi and Malik, 2000), to generate a dichotomy of the samples during each iteration. Approximate Normalized Cut avoids the pitfalls of the more usual Minimum Cut approach, which tends to produce highly unbalanced partitions. Moreover, there is an efficient algorithm for Approximate Normalized Cut, which makes it an ideal module in a iterative algorithm. For the feature selection part, we use a mixture of feature evaluation experts based on independent feature modeling, information gain ranking, and Markov blanket filtering (Koller and Sahami, 1996) to remove non-discriminative, irrelevant and redundant genes, respectively, from the original gene set.

As an example we demonstrate the performance of our algorithm on a two-way clustering problem (a generalization to multi-way clustering is straightforward) that partitions leukemia samples (consisting of two subtypes) into two groups based on their gene expression profiles.

The remainder of this paper is structured as follows: Section 2 describes the approximate normalized cut algorithm we used for object partitioning. Section 3 then describes the three types of feature selection techniques we used for our feature filtering system. In Section 4 we present the full CLIFF algorithm. Section 5 goes on to describe our experimental results and in Section 6 we conclude with brief summary and discussion of our algorithm.

THE CLUSTERING ALGORITHM

Preliminaries

Consider clustering based on a similarity measure between objects. We represent the set of objects to be partitioned

as the vertex set $\mathbf{V}$ of a complete graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$. Associated with each object i is an expression vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ in a m-dimensional feature space. Each edge $e_{i,j} \in \mathbf{E}$ has a weight w_{ij} corresponding to the degree of similarity between objects i and j.

Intuitively, a good two-way partition of the graph should have the property that the sum of the weights of the edges joining the two subgraphs is small. Accordingly, several microarray clustering approaches, such as the recently reported CLICK (Shamir and Sharan, 2000) algorithm, partition the graph using a minimum-cut algorithm, which minimizes the sum of the weights of the edges joining the two parts. However, the weakness of this approach is that there is little guarantee that the algorithm will not go astray and generate partitions that are highly unbalanced, and thus sophisticated pruning techniques need to be developed to explicitly enforce cut balance. The Approximate Normalized Cut (NCut) algorithm, which was recently applied to image segmentation by (Shi and Malik, 2000), avoids the unbalanced cut difficulty in a natural and efficient way.

Normalized Cut

For any two (not necessarily disjoint) subsets $\mathbf{A}$ and $\mathbf{B}$ of the vertex set $\mathbf{V}$, define $w(A, B) = \sum_{u \in A} \sum_{v \in B} w(u, v)$. A minimum cut is a partition of the vertex set into two subsets, $\mathbf{A}$ and $\bar{\mathbf{A}}$, which minimizes $w(A, \bar{A})$. By contrast, in the Normalized Cut framework, we normalize $w(A, \bar{A})$ by scaling it relative to $w(A, V)$ and to $w(\bar{A}, V)$, where, for example, $w(A, V)$ is the sum, over all vertices $v \in \mathbf{A}$, of the total weight of the edges incident with v. This scaling eliminates the bias toward highly unbalanced cuts. Specifically, we define the *normalized weight* of the cut $A, \bar{A}$ as follows:

$$Ncut(A, \bar{A}) = \frac{w(A, \bar{A})}{w(A, V)} + \frac{w(A, \bar{A})}{w(\bar{A}, V)}. \tag{1}$$

An *optimal normalized cut* is a cut of minimum normalized weight.

Unfortunately, computing an optimal normalized cut is NP hard even if all edge weights are non-negative. In an effort to efficiently compute a cut of approximately minimum normalized weight, (Shi and Malik, 2000) reformulated this problem using a linear algebra notation and showed that the problem of computing an optimal normalized cut can be formulated as follows:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W)\mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \tag{2}$$

$$\text{subject to: } \mathbf{y} \in \{1, -b\}^n, \tag{3}$$

$$\text{and } \mathbf{y}^T \mathbf{D1} = 0. \tag{4}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{D}$ is a diagonal matrix with $d(i, i) = \sum_j w(i, j)$, and b is a positive constant.

There is a one-to-one correspondence between cuts and feasible solutions $\mathbf{y}$ for (2), such that the value of the objective function at $\mathbf{y}$ is equal to the normalized weight of the corresponding cut. Note that the objective function of (2) is the *Rayleigh Quotient*. If we relax constraint (3) and allow elements in $\mathbf{y}$ to take on any real values, by the *Rayleigh Quotient Theorem*. we can minimize $Ncut$ by solving the generalized eigenvalue system:

$$(\mathbf{D} - \mathbf{W})\mathbf{z} = \lambda \mathbf{D}\mathbf{z} \tag{5}$$

It can be shown that the eigenvector (denoted as $\mathbf{z}^*$) associated with the second-smallest eigenvalue of the generalized eigenvalue system is the optimal solution to this relaxed problem.

The graph partition corresponding to the approximate solution to (2) can be recovered by choosing the best of n possible partitions, each of which corresponds to separating the large components of $\mathbf{z}^*$ from the small components. For each threshold k, $1 \leq k \leq n$, let:

$$\mathbf{A}_k = \{i \mid z_i^* \text{ among } k \text{ largest elements of } z^*\}$$
$$\mathbf{B}_k = \{i \mid z_i^* \text{ among } n - k \text{ smallest elements of } z^*\}$$

The Approximate Normalized Cut algorithm selects $\{A_{k^*}, B_{k^*}\}$, where k^* is the best of the n thresholds according to the normalized cut criterion:

$$k^* = \arg\max_k Ncut(\mathbf{A}_k, \mathbf{B}_k).$$

It is easy to generalize the Approximate Normalized Cut algorithm from 2-way partitioning to multi-way clustering. One way is to perform recursive 2-way cuts until each resulting subset of the vertices is a singleton. Essentially, this produces a binary hierarchy in a top-down direction, and the cut weight associated with each branching point in the hierarchy reflects the degree of dissociation of the two subtrees directly below the branching point. Given this hierarchy, one can choose a clustering according to the desired degree of granularity, with each cluster corresponding to a subtree. Another way is to use several eigenvectors of the generalized eigenvalue system to come up with a simultaneous K-way cut (see (Shi and Malik, 2000) for details).

Weight definition and difficulties with similarity-based clustering approach

The quality of clustering using Normalized Cut or any other algorithm based on pairwise similarities fundamentally depends on the weights - the w_{ij}'s - that are provided as input. There are many ways to define a similarity measure for biological objects (Szallasi and Somogyi, 2001).

In this paper, we use Pearson correlation coefficient (under an exponential kernel) between the expression vector $\mathbf{x}_i$ and $\mathbf{x}_j$ (denoted as $\rho(\mathbf{x}_i, \mathbf{x}_j)$) to capture the similarity between objects i and j:

$$w_{ij} = \exp\{-\frac{(1-\rho(\mathbf{x}_i, \mathbf{x}_j))}{\sigma}\}, \quad (6)$$

where σ is a scaling factor controlling the sensitivity of clustering to the 'strength' of the pairwise similarity. We call the resulting $\mathbf{W}$ matrix the *affinity matrix*.

A common problem with this pairwise similarity based approach is that the resolution of the similarity measure does not scale well with the dimensionality of the feature space, especially when the measurements are noisy, or the majority of the features are irrelevant to the target partition, or multiple meaningful partitions are possible. As we will show later, by doing feature selection, we can effectively reduce the clustering problem to a lower dimensional space, where the pairwise similarities between *mates* (objects from the same group) and *non-mates* (objects from different groups) resolve into two distinguishable distributions. With this enhancement, we can significantly improve the graph-theoretic clustering and are better able to perform probabilistic modeling in the resulting feature subspace.

THE MIXTURE OF FEATURE SELECTION EXPERTS

Preliminary

Among the many thousands of genes simultaneously measured in a microarray experiment, it is unlikely that all of their expressions are related to a particular partition of the samples. In the analysis of a biological system, the following 'rules of thumb' regarding gene functions are often assumed. 1) A gene can be in either the 'on' or 'off' state (or maybe more subtly, 'up', 'neutral' or 'down'); 2) not all genes simultaneously respond to a single physiological event; 3) gene functions are highly redundant.

If we have complete knowledge of the gene regulatory network *a priori*, we can just neglect the non-discriminative, irrelevant and redundant genes explicitly, and work in a lower dimensional feature space. Otherwise, one needs to probe the relevance of different subsets of the genes to the target biological event (which often induces a partition) in order to choose the best feature subspace. Exhaustive search of the power set of the feature set is intractable when the number of features is large, as in the case of microarray data. Various heuristic feature selection methods have been developed. Based on the aforementioned assumptions about gene activity, we use the following three approaches to probe the relevance of each feature in a sequential fashion: 1) independent feature modeling (unsupervised); 2) information gain ranking (supervised); 3) Markov blanket filtering (supervised).

Independent feature modeling

An important empirical assumption about the activity of genes, and hence their expression, is that they generally assume a few distinct biological states (*e.g.* either 'on' or 'off'). The combination of such discrete patterns from multiple genes determines the sample phenotype. Given this assumption, we expect that the marginal probability of measurements over each individual gene can be modeled as a univariate mixture with (say) two components (which includes the degenerate case of a single component). If a gene has all its measurement points in a single state, or in two states for which the probability distributions overlap heavily, we may conclude that the gene probably contributes little to discriminating the samples.

Formally, for each feature F_i, we have a vector $\mathbf{f}_i = \{x_{1i}, \ldots, x_{Ni}\}$ of measurements over the N samples. Assuming that F_i can be in either the 'on' or 'off' state in each sample n, and that the measurements in a given state come form a Gaussian distribution, we can model the likelihood of $\mathbf{f}_i$ as a univariate mixture of two Gaussians (easily generalizable to k components):

$$p_c(\mathbf{f}_i|\theta_i) = \prod_{n=1}^{N}\prod_{k=0}^{1}\left(\pi_{i,k}\left[\frac{1}{\sqrt{2\pi}\sigma_{i,k}}\exp\left\{-\frac{(x_{ni}-\mu_{i,k})^2}{2(\sigma_{i,k})^2}\right\}\right]\right)^{z_n^k} \quad (7)$$

Here $\theta_i = (\theta_{i,0}, \theta_{i,1})$, where $\theta_{i,k} = (\pi_{i,k}, \mu_{i,k}, \sigma_{i,k})$, and z_n^k is 1 if sample n has state k, and 0 otherwise.

Learning this mixture model is easy using EM. From the model we can derive a classification hypothesis $h(x)$ as to the state from which feature value x is drawn: $h(x) = 1$ if $\pi_{i,1}P(x|\theta_{i,1}) \geq \pi_{i,0}P(x|\theta_{i,0})$ and 0 otherwise.

The Bayes error (probability of misclassifying a sample drawn from the mixture of Gaussians) is:

$$\epsilon_{Bayes} = \pi_0 P(h(x) = 1|z_x = 0) + \pi_1 P(h(x) = 0|z_x = 1); \quad (8)$$

This error is the best we can achieve when classifying samples using F_i only, and is intuitively a reasonable measure of the discriminability of F_i. For highly discriminative features, we anticipate that ϵ_{Bayes} is small. Thus we may rank features according to their discriminability, in an order determined by ϵ_{Bayes}.

Note that a mixture model can also be used as a quantizer, allowing discretization of the measurements for a given feature. We simply replace the measurement x_i with the associated binary value $f_i = h(x_i)$ (such quantization scheme is adopted throughout the rest of

the feature selection sections in this paper unless otherwise specified). This discretization allows us to bring information-theoretic techniques to bear in determining the degree of agreement between a feature and any given partition of the samples. We can also reuse the same quantization in any further partitions of the samples such as in the case of hierarchical clustering.

Information gain ranking

As mentioned before, in the supervised learning paradigm, feature quality is much easier to assess, because we can explicitly measure the degree of agreement of each feature to the reference sample partition. A standard measure for such purpose is the *information gain*. For a *reference partition* $S_1, \ldots, S_C$, let the probability of each part be the empirical proportion: $P(S_c) = |S_c|/|S|$. Now suppose a test on feature F_i induces a partition of the training set into $E_1, \ldots, E_K$. Let $P(S_c|E_k) = P(S_c \cap E_k)/P(E_k)$. The information gain due to this feature with respect to the original partition is:

$$I_{gain} = H(P(S_1), \ldots, P(S_C)) - \sum_{k=1}^{K} P(E_k) H(P(S_1|E_k), \ldots, P(S_C|E_k)), \quad (9)$$

where H is the entropy function (the entropy of a discrete probability distribution $\{p_i\}$ is defined as $-\sum_i p_i \ln p_i$).

The information gain measure is applicable when the reference partition is consistent with the target concept we would like to learn. For simplicity, we deal with a 2-way partition with the two parts denoted S_0 and S_1. Note that we also need a decision rule for each feature F_i in order to generate the partition induced by that feature. For this, we turn to the classification rule $f_i = h(x_i)$ obtained from independent feature modeling described in last section. This induces a subjective partition based on measurements on F_i only. Naturally, higher information gain for F_i suggests that the F_i-induced partition is more consistent with the reference partition, and thus F_i is a relevant feature.

Markov blanket filtering

It is natural to assume that many features, such as gene transcription levels, are redundant or secondary responses to the biological or experimental conditions that distinguish the different samples. It is often desirable to retain only the non-redundant, directly relevant features. We can formalize our goal as follows: select a feature subset $\mathbf{G} \subseteq \mathbf{F}$, such that, the two distributions $P(C|\mathbf{F} = \mathbf{f})$ and $P(C|\mathbf{G} = \mathbf{f}_G)$ are the same or very close, where $\mathbf{f}_G$ is just the projection of $\mathbf{f}$ onto the variables in $\mathbf{G}$.

For some computational purposes it will be convenient to replace the original feature values by the binary values that result from discretization using a mixture of two Gaussians. This replacement is done throughout this subsection except where otherwise noted. Thus, for example, f_i denotes the binary value resulting from discretization of the original real-valued measurement x_i. Define the distance of a feature subset $\mathbf{G} \subseteq \mathbf{F}$ to $\mathbf{F}$, as the expectation, over discrete values $\mathbf{f}$ of the feature subset $\mathbf{F}$, of the cross-entropy (denoted as $D(. \,\|.)$) between the conditional distribution of C given $\mathbf{f}$ and the conditional distribution of C given $\mathbf{f}_G$, the projection of $\mathbf{f}$ on G:

$$\Delta_{\mathbf{G}} = \sum_{\mathbf{f}} P(\mathbf{f}) \cdot D(P(C|\mathbf{F} = \mathbf{f}) \,\|\, P(C|\mathbf{G} = f_{\mathbf{G}})),$$

We want to find a small feature set $\mathbf{G}$ for which $\Delta_{\mathbf{G}}$ is small. Intuitively, if the information contributed by a feature F_i is subsumed by some small subset of the other features (which is often called a *Markov blanket* of F_i in the graphical models literature), then we should be able to neglect F_i without compromising the accuracy of class prediction. This naturally suggests a filtering approach of deleting 'bad' features one by one, rather than conducting a combinatorial search through the power set of the feature set. (Koller and Sahami, 1996) proposed the technique of sequentially identifying such features based on the (non)existence of a Markov blanket in the candidate feature set. In most cases, however, few if any features will have a Markov blanket of limited size, and we must instead look for features that have an "approximate Markov blanket".

For this purpose we define:

$$\Delta(F_i|\mathbf{M}_i) = \sum_{f_{\mathbf{M}_i}, f_i} P(\mathbf{M}_i = f_{\mathbf{M}_i}, F_i = f_i) \, D(P(C|\mathbf{M}_i = f_{M_i}, F_i = f_i) \,\|\, P(C|\mathbf{M}_i = f_{M_i})) \quad (10)$$

If $\mathbf{M}_i$ is a Markov blanket for F_i then $\Delta(F_i|\mathbf{M}_i) = 0$. Since this fortunate case is unlikely to occur, we relax the condition and seek a set $\mathbf{M}_i$ such that $\Delta(F_i|\mathbf{M}_i)$ is small. Since the goal is to find a small irredundant feature subset, and those features that form an approximate Markov blanket of feature F_i are most likely to be more strongly correlated to F_i, we construct a candidate Markov blanket for F_i by collecting the k features that have the highest correlations with F_i, where k is a small integer. In computing these correlations we use the original real values of the features, rather than the discretized values. Here is how the algorithm goes as proposed in (Koller and Sahami, 1996):

Initialize
- $\mathbf{G} = \mathbf{F}$

Iterate
- For each feature $F_i \in \mathbf{G}$, let $\mathbf{M}_i$ be the set of k features $F_j \in \mathbf{G} - \{F_i\}$ for which the correlations between F_i and F_j are the highest.

- Compute $\Delta(F_i|\mathbf{M}_i)$ for each i
- Choose the i that minimizes $\Delta(F_i|\mathbf{M}_i)$, and define $\mathbf{G} = \mathbf{G} - \{F_i\}$

This heuristic sequential method is far more efficient than methods that conduct a more extensive search over subsets of the feature set. The heuristic method only requires independent feature modeling to discretize (or binarize, in this case) the data for each gene, followed by computation of quantities of the form $P(C|\mathbf{M}_i = \mathbf{f}_{M_i}, F_i = \mathbf{f}_i)$ and $P(C|\mathbf{M}_i = \mathbf{f}_{M_i})$.

THE FULL ALGORITHM

Thus far we have described two types of modules for the analysis of gene expression data. The Approximate Normalized Cut (NCut) algorithm takes an affinity matrix defined over objects in a certain feature space as input, and outputs a partition of the objects. The mixture of feature selection experts, except for the first feature modeling stage, where no supervised information is needed, take a given partition as reference, and output an ordering of all the features in terms of their relevance or irredundancy with respect to the reference partition. Here we outline an procedure that combines these two modules in an interactive way, alternating between computing a new reference partition given the currently selected features, and selecting a new set of features based on the current reference partition. The bootstrapping step to select an initial set of features is based entirely on independent feature modeling.

Specifically, we first use the unsupervised independent feature modeling technique to rank all features in terms of their discriminability. Then we generate an initial partition based on the k most discriminative features, where k is specified in advance. Based on this partition, we can treat feature selection roughly as a 'supervised' learning problem, where information gain ranking and Markov blanket filtering can be applied, and the newly determined feature subset can then be used to generate a new partition, which in turn can be used to further improve the feature selection. The scenario is that although we do not know the exact target partition *a priori*, with respect to which we would like to optimize the feature subset, at each iteration we can expect to obtain an approximate partition that is close to the target one, and thus allows the selection of an approximately good feature subset, which will hopefully draw the partition even closer to the target partition in the next iteration. The algorithm is similar in spirit to the EM algorithm (Dempster et al., 1977), where one searches the parameter space for local minima via a coordinate descent type of approach (improving an objective function along one direction at a time, assuming invariance along other directions). Such an approach will always lead to some local minimum, either a point or a basin (a cyclic set of attraction points). Figure 1 lists the details of the algorithm.

Initialize
- using independent feature modeling, compute the Bayes error ϵ_{Bayes} for each feature
- $\mathbf{G}_1$ = {features corresponding to the k smallest Bayes errors }
- generate partition C_1 based on $\mathbf{G}_1$ using NCut, let $C = C_1$

Iterate
- using C as a reference partition, compute the information gain I_{gain} for each feature
- $\mathbf{G}_2$ = {features corresponding to the k smallest I_{gain} values }
- generate partition C_2 based on $\mathbf{G}_2$ using NCut, let $C = C_2$

- order features according to $\Delta(F_i|\mathbf{M})$ via Markov blanket filtering using C as reference
- $\mathbf{G}_3$ = {features corresponding to the k smallest values of $\Delta(F_i|\mathbf{M})$ }
- generate partition C_3 based on $\mathbf{G}_3$ using NCut, let $C = C_3$

if C and $\mathbf{G}_3$ converge, **end** iteration

Fig. 1. The CLIFF algorithm.

EXPERIMENTS AND RESULTS

In this section, we report the results of using CLIFF on a microarray clustering problem. Our data is a collection of 72 Leukemia patient samples reported in (Golub et al., 1999). Each sample is measured over 7130 genes. According to pathological/histological criteria, these samples include 47 type I leukemias (called ALL) and 25 type II leukemias (called AML). We want to see whether CLIFF is able to generate a partition that matches well with this pathological/histological categorization of the samples based on their gene expression profiles.

Feature relevance analysis of the expression data

As a prelude to partitioning the samples without the help of a reference partition, we used the 'correct' partition (the actual subtype labeling of the Leukemia samples) to gauge the possible value of feature selection. We evaluated each gene according to three measures:

1. Its degree of discriminability, as measured by the Bayes error of a mixture of two Gaussians model;
2. Its information gain with reference to the correct reference partition;
3. The value of $\Delta(F_i|\mathbf{M})$, where a small value indicates that the feature is redundant.

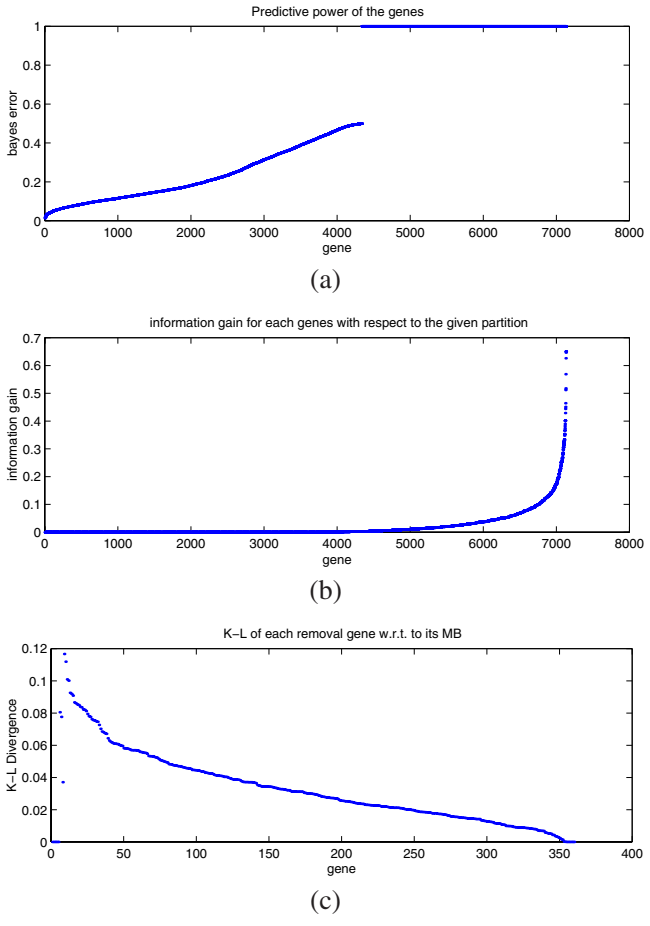

Fig. 2. Feature selection using using a 3-stage procedure. (a) Genes ranked by ϵ_{Bayes} (Eq. 8), which indicates discrimination power. (b) Genes ranked by I_{gain} (Eq. 9), which indicates degree of relevance. (c) Genes ranked by $\Delta(F_i|\mathbf{M})$ (Eq. 10) which indicates degree of redundancy.

We found that there was great variation among the genes with respect to each of these measures.

Figure 2a shows the Bayes errors (defined by Eq. 8) of the genes in ascending order. It can be seen that, only a small percentage of the genes actually resolve into two states with an error rate significantly better than a random guess ($\epsilon \ll 0.5$). Not all the gene expressions can be successfully modeled as a mixture of two Gaussians in practice, for example, due to presence of outliers or possibly multiple (rather 2) underling states (we spare further discussion of this issue for the sake of simplicity).

Figure 2b plots the information gain of the individual genes with respect to the reference partition. Note that only a very small fraction of the genes induce a significant information gain, and hence are indeed relevant. We take the top 360 genes from this list (rather than the whole set, for the sake of computational efficiency) to proceed with the (approximate) Markov blanket filtering.

Figure 2c shows the value of $\Delta(F_i|\mathbf{M}_i)$ (defined by Eq. 10) for each F_i, which measures to what extent $\mathbf{M}_i$ subsumes information carried by F_i, and thus renders F_i redundant. Genes are removed in increasing order of $\Delta(F_i|\mathbf{M}_i)$, since a small value of this quantity indicates more complete blanketing of F_i by $\mathbf{M}_i$. In our experiment we choose the size of each Markov blanket to be small to avoid fragmenting our small sample set too much. Since in a real biological regulatory network system each gene is expected to be directly influenced by only a few others, our small Markov Blanket assumption is plausible.

In a separate paper (Xing et al., 2001), we used the same mixture of feature selection experts used in this paper in a supervised classification setting. We found that when only a small subset of selected features are used for concept learning, the resulting classifier significantly outperforms the one that takes the full feature set into account. Thus, it is reasonable to expect that in a cluster analysis, feature selection will lead to a less error-prone result.

Clustering via CLIFF

We compared four algorithms for partitioning the Leukemia samples into two classes:

1. Approximate NCut without feature selection;
2. K-means (for K =2) without feature selection;
3. CLIFF (Approximate NCut with iterative feature selection and partitioning);
4. K-means (for K = 2) with feature selection.

We used the quality measures applied in (Shamir and Sharan, 2000), the *Minkowski measure* (M) and the *Homogeneity* (H_{Ave} and H_{Min}) to evaluate the quality of the resulting partitions. The *Minkowski measure* of disagreement between the computed partition and the correct partition is defined as $\sqrt{A/B}$ where A is the number of pairs of samples that are in the same part of one, but not both, of the two partitions, and B is the number of pairs that lie in the same part of the true partition. The quantities H_{Ave} and H_{Min} refer only to the computed partition, not to the true partition; they are the average and minimum of the correlations between the expression vector of a sample and the mean of the expression vectors of the samples in the same cluster.

Figure 3 shows the results from Approximate NCut and CLIFF. From the graphic display of the input affinity matrix $\mathbf{M}$ (upper left panel), we can see that, without feature selection (Figure 3a), the contrast of strong pairwise affinity to poor pairwise affinity is very low and the eigenvector on which NCut bases its partition does not exhibit a sharp separation between small and large values. This lack of separation seems to be due to the large number of

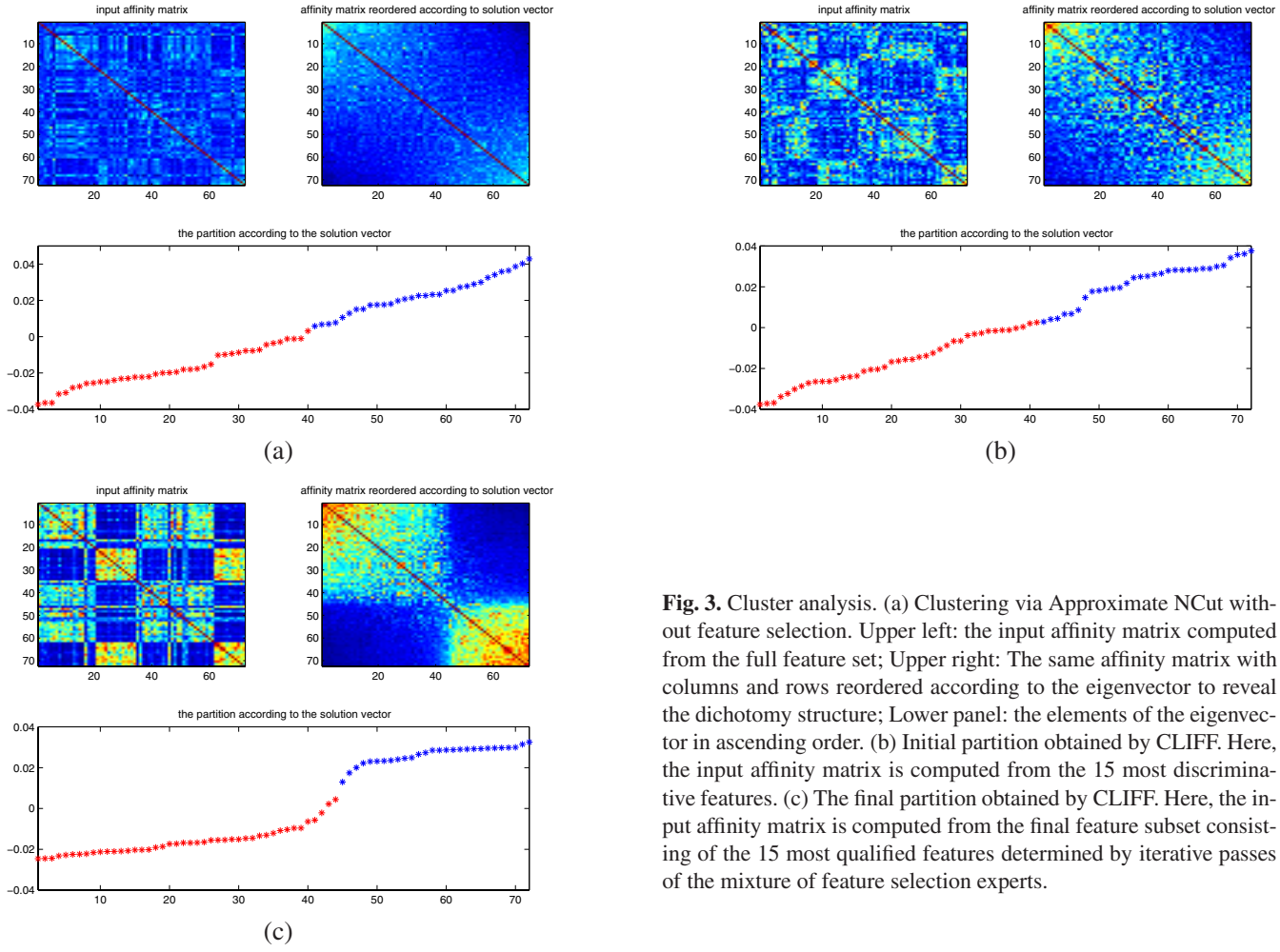

Fig. 3. Cluster analysis. (a) Clustering via Approximate NCut without feature selection. Upper left: the input affinity matrix computed from the full feature set; Upper right: The same affinity matrix with columns and rows reordered according to the eigenvector to reveal the dichotomy structure; Lower panel: the elements of the eigenvector in ascending order. (b) Initial partition obtained by CLIFF. Here, the input affinity matrix is computed from the 15 most discriminative features. (c) The final partition obtained by CLIFF. Here, the input affinity matrix is computed from the final feature subset consisting of the 15 most qualified features determined by iterative passes of the mixture of feature selection experts.

irrelevant features. NCut without feature selection generates a partition that significantly disagrees with the original Leukemia subtype labeling of the samples (Cluster I contains 7 AML and 33 ALL, and cluster II has 18 AML and 14 ALL).

Table 1. Performance comparison of clustering algorithms with and without iterative feature selection.

	no feature selection			with feature selection		
algorithm	M	H_{Ave}	H_{Min}	M	H_{Ave}	H_{Min}
K-Means	0.950	0.164	-0.097	0.903	0.674	-0.297
N-Cut	0.938	0.337	0.057	0.387	0.633	-0.073

* N-cut with iterative feature selection is just CLIFF

The number of features selected and used to compute the affinity matrix **M** during each iteration is chosen empirically, and the clustering result is sensitive to different choices of this number. For some other choices CLIFF identified other tight and well separated dichotomies associated with a small set of relevant genes, or alternated between 2 or 3 such dichotomies. It is possible that such dichotomies may be genetically meaningful even though they are not revealed or documented in the pathological sample records.

When we use a small number of features determined by the feature selection filters, the affinity matrix exhibits much more contrast among the affinity strengths of different pairs of samples (Figure 3b and 3c). The initial partition derived from the 15 most discriminative features (later on we use the 20 best features determined by the feature filters for the iterative clustering) is already somewhat close to the actual labeling of the samples (Cluster I: AML/ALL=1/40, Cluster II: AML/ALL=24/7). It takes 9 full iterations for CLIFF to converge to an invariant clustering solution. In the final partition, cluster I contains 44 samples, all of which are ALL; cluster II contains 28 samples, of which 25 are AML and 3 are ALL. Given that no expert knowledge is provided regarding which genes out of the more than 7000 candidates are relevant to the leukemia subtype categorization, the degree

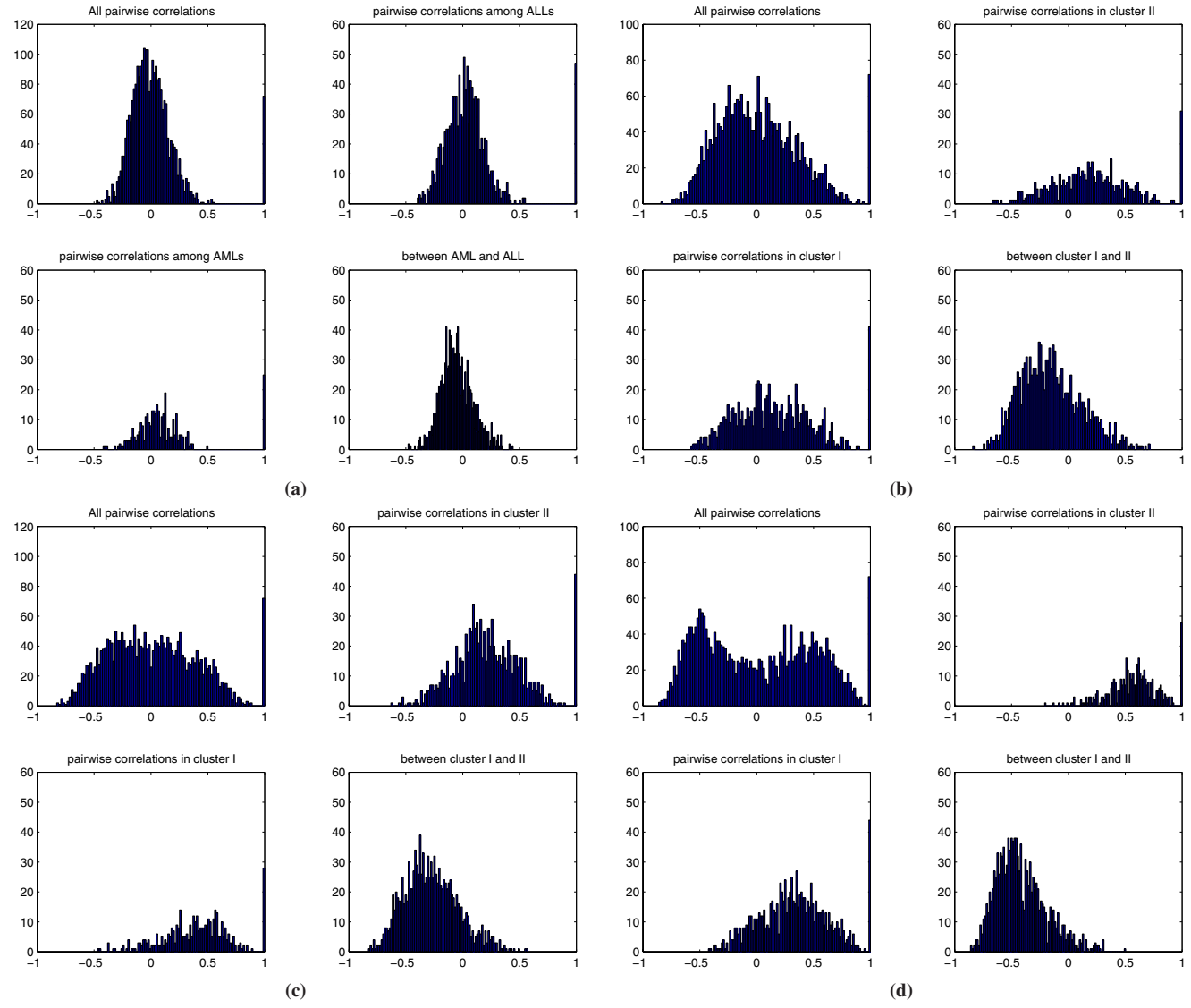

Fig. 4. Change of sample affinity distributions during the CLIFF iterations. (a) Histograms of pairwise sample correlations in the full feature space. (b) Correlations in the space spanned by the 15 most discriminative features. (c) Correlations in the space of features determined in the first full iteration of CLIFF. (d) Correlations in the final feature space determined by CLIFF.

of agreement of the partition generated by CLIFF to the actual leukemia sub-categorization is remarkable. It should be noted, however, that other phenotypically significant partitions that are unrelated to leukemia may still be left undetected by our algorithm.

Detailed analyses not given here also showed that the Markov Blanket filter imposes more influence on the stability and correctness of the clustering than the information gain filter, especially when the number of the features to be finally used is small.

In a comparison of the clustering result using different approaches (Table 1), we can see that CLIFF outperforms both K-means with feature selection and NCut without feature selection.

Recall that the CLICK algorithm (Shamir and Sharan, 2000) assumes that the distributions of inter- and intra-cluster pairwise sample correlations are well separated and Gaussian. As shown in Figure 4 these assumptions do not hold when all the original features are used. However, we observed that, as the CLIFF iterations proceeded and only the most significant features were retained, there was a clear trend of gradual separation of the two distributions. By the end of the process the two distributions appear to be reasonably separable, although not exactly Gaussian.

This observation suggests that, for some data sets, the assumptions underlying CLICK may be valid only when relevant and irredundant features, rather than all the features, are considered in sample affinity measure.

It is worth mentioning that ten of the top 20 genes (7 of the top 10 genes) of our final feature ranking are among the 50 'informative' genes used in (Golub et al., 1999) (which is scored by some 'relevance' measure). But since our procedure filters out redundant genes, it is not surprising that many of the genes in that list are not included here. The affinity matrix derived from the top 20 genes shows a very strong contrast among the affinity strengths, and the eigenvector from which NCut derives its final partition has a sharp distinction between the values in the two parts of the final partition.

CONCLUSION

In this paper we presented the use of CLIFF: Clustering via Iterative Feature Filtering, in the cluster analysis of high-dimensional gene expression data, for which the presence of large numbers of irrelevant and redundant features and the limited number of samples often prevent accurate grouping of the samples. Our results show that even without sample labels as training information, it is still possible to do feature selection together with cluster analysis by coupling the two processes in a such way that each process uses the output of the other process as an approximate input. The dimensionality reduction achieved in our algorithm should enhance the performance of any clustering algorithm, not only the eigenvalue-based NCut algorithm that we used. CLIFF is fully generalizable to arbitrary multi-way clustering, either through recursive 2-way cuts or simultaneous use of several eigenvectors. As with the CLICK algorithm, CLIFF needs no prior assumptions on the structure or the number of the clusters as long as a proper threshold measure of partition quality is given. Furthermore, the complex adoption and merging steps in CLICK for dealing with singletons and unbalanced cuts are avoided by the Normalized Cut technique, and the assumption of a mixture of two Gaussians for distributions of the similarity measures (which may not be true in a high-dimensional feature space) is alleviated due to the iterative feature selection techniques.

In summary, our results suggest that the CLIFF algorithm, with its iterative use of a mixture of feature filters followed by reclustering, is capable of capturing the partition that characterizes the samples but is masked in the original high-dimensional feature space. Not only can hidden biologically meaningful partitions of the sample set be identified in this way, but also the selected features are of significant interest because they represent a set of causal factors that elicit such partitions. Such information can be used to establish causal models connecting quantitative microscopic features (gene expression patterns) with qualitative and empirical macroscopic phenotypes such as disease symptoms and pathologies, and can serve as a basis for grouping genes into functional clusters.

REFERENCES

Alter, O., P. Brown, and D. Botstein (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci USA 97*, 10101–10106.

Dempster, A., N. Laird, and M. Revow (1977). Maximum likelihood from implete data via the em algorithm. *Journal of the Royal Statistical Society B 39(1)*, 1–38.

Eisen, M., P. Spellman, P. Brown, and D. Bottstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA 95*, 14863–14868.

Golub, T., S. D.K., P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. L. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science 286*, 531–537.

Hastie, T., R. Tibshirani, M. Eisen, P. Brown, D. Ross, U. Scherf, J. Weinstein, A. Alizadeh, L. Staudt, and D. Botstein (2000). Gene shaving: a new class of clustering methods for expression arrays. In *Tech. report, Stanford University*.

Kohavi, R. and G. John (1997). Wrapper for feature subset selection. *Artificial Intelligence 97*, 273–324.

Koller, D. and M. Sahami (1996). Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI Press.

Shamir, R. and R. Sharan (2000). Click: A clustering algorithm for gene expression analysis. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*. AAAI Press.

Shi, J. and J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8)*, 888–905.

Szallasi, Z. and R. Somogyi (2001). Genetic network analysis - the millennium opening version. In *Pacific Symposium of BioComputing Tutorial*.

Vapnik, V. and A. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications 16(2)*, 264–280.

Wen, X., S. Fuhrman, G. Michaels, D. Carr, S. Smith, J. Barker, and R. Somogyi (1998). Large-scale temporal gene expression mapping of cns development. *Proc Natl Acad Sci USA 95*, 334–339.

Xing, E., M. Jordan, and R. Karp (2001). Feature selection for high-dimensional genomic microarray data. In *the Eighteenth International Conference on Machine Learning, in press*.