An Iterative Strategy for Pattern Discovery in High-dimensional Data Sets

Chun Tang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
chuntang@cse.buffalo.edu

Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
azhang@cse.buffalo.edu

ABSTRACT

High-dimensional data representation in which each data item (termed target object) is described by many features, is a necessary component of many applications. For example, in DNA microarrays, each sample (target object) is represented by thousands of genes as features. Pattern discovery of target objects presents interesting but also very challenging problems. The data sets are typically not task-specific, many features are irrelevant or redundant and should be pruned out or filtered for the purpose of classifying target objects to find empirical pattern. Uncertainty about which features are relevant makes it difficult to construct an informative feature space. This paper proposes an iterative strategy for pattern discovery in high-dimensional data sets. In this approach, the iterative process consists of two interactive components: discovering patterns within target objects and pruning irrelevant features. The performance of the proposed method with various real data sets is also illustrated.

Categories and Subject Descriptors

I.5.3 [**Computing Methodologies**]: Pattern Recognition—*Clustering*

General Terms

Algorithms, Design

Keywords

iterative, empirical pattern, microarray, unsupervise, feature

1. INTRODUCTION

High-dimensional data representation, in which each data item is described by many features is a necessary component of many applications. For example, customer records compiled by banks or supermarkets contain many features to represent each customer. In biological systems, the DNA microarray technology permits rapid, large-scale screening for patterns of gene expression and gives simultaneous, semiquantitative readouts on the level of expression of thousands of genes for samples [7, 12, 28]. In that instance, each sample is represented by its features in the form of thousands of genes. In this paper, we will refer to entities such as samples as *target objects*.

In the scenarios mentioned above, clustering or classification is needed on both the level of features and of target objects. Many data-mining approaches have been proposed which cluster or group related features pertaining to supermarket customers so their shopping patterns can be determined [17]. With gene expression data, researchers are interested in identifying genes with interaction or having similar cellular functions [3]. Additionally, classification of customers or samples is also useful. For example, classification methods have been used to group different types of credit-card holders. Biological samples may be classified into homogeneous groups which may correspond to particular macroscopic phenotype, such as clinical syndromes or cancer types [16]. In this instance, samples are viewed as the objects to be classified, with the level of expression of each gene playing the role of the features. Classifying samples or customers to reveal their empirical types is also regarded as the process of pattern discovery in the target objects.

Pattern discovery of target objects presents interesting but also very challenging problems. The data sets are typically not task-specific, the number of target objects is limited but feature dimension is very large, many features are irrelevant or redundant for the purpose of target object classification and should be pruned out or filtered. Uncertainty about which features are relevant makes it difficult to construct an informative feature space.

1.1 Related work

The existing target data classification methods fall into two major categories: supervised approaches and unsupervised approaches. The goal of supervised approaches is to build a classifier to predict the labels for future coming objects. A variety of supervised clustering methods have been proposed, such as neighborhood analysis [16], the support vector machine [5, 14], statistical approaches such as the maximum-entropy model [19], SAM (significance analysis of

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

microarrays)[33], and ranking-based methods [4, 21, 24].

In this paper, we will focus on unsupervised approaches which assume little or no prior knowledge. The goal of such approaches is to partition the dataset into statistically meaningful classes [4]. The hierarchical clustering (HC) [11] and K-means clustering algorithms [22, 31], as well as self-organizing maps (SOM) [20, 30], are the major unsupervised clustering methods which have been commonly applied to various data sets. Methods using principal component analysis (PCA) [36] have been proposed to reduce the number of features involved in the data sets. However, since the results largely depend on the data distribution, such approaches do not necessarily fully capture most of the cluster structures. For biological applications, Alon at al. [3] proposed a stimulated annealing method in which genes and samples were clustered independently. Getz et al. [15] proposed a coupled two-way clustering method to identify subsets of both genes and samples. Xing et al. [34] and Ding et al. [9] proposed clustering methods which iteratively use sample partitions as a reference to filter genes. None of these approaches offer a definitive solution to the fundamental challenge of detecting meaningful patterns in target objects while pruning out irrelevant features in a context where little domain knowledge is available.

1.2 Contribution of the present research

In this paper, we will present a new iterative strategy for target pattern discovery in high-dimensional data sets. In this approach, the iterative process consists of two interactive components: discovering patterns within target objects and excising irrelevant feature information. Using these two processes, we can delineate the relationships between feature groups and target-data groups while conducting an iterative search for target patterns and detecting significant features of empirical interest. The performance of the proposed method will be illustrated in the context of various real data sets.

The remainder of this paper is organized as follows. Section 2 introduces the concepts underlying this iterative strategy while the algorithm to implement these concepts is presented in Section 3. Experimental results appear in Section 4 and concluding remarks in Section 5.

2. STRATEGY AND CONCEPTS

In this section, we will introduce the concepts underlying the proposed strategy for pattern discovery in high dimensional data sets. In this concept, target objects are usually represented as points in a high dimensional space, where each dimension represents a distinct feature describing the data. The strategy presented is intended to simplify the discovery of patterns in the data represented in very large feature spaces.

The goal involves two interrelated tasks: detection of meaningful patterns within the target objects and selection of those significant features which contribute to the empirical patterns of target data. This is different from subspace clustering [2, 1, 6, 35] which is to find a set of subspace clusters and the corresponding feature subsets for each cluster such that the objects in one subspace cluster are close to each other in the subspace defined by these features. In subspace clustering, the feature subsets for different clusters are different while our goal is to find a single set of features to classify all objects. Furthermore, subspace clustering approaches are suitable for such data sets that object number is very large, but feature dimension is relatively small (e.g. $10^4 \sim 10^5$ objects versus $20 \sim 100$ features in [2, 1, 6, 35]). But our research is analyzing some real data sets that the number of target objects is limited but feature dimension is very large. For example, in microarray data sets, the number of samples is usually less than 100 due to the cost of the experiments, but the number of genes (regarded as features for samples) is around $10^3 \sim 10^4$. Within such kind of data sets, the factors such as the sparsity of data, the high dimensionality of the feature space, and the high percentage of irrelevant or redundant features make it very difficult either to classify target objects or pick out substantial features with unsupervised manner.

To address these problems, we propose the unsupervised iterative pattern-discovery strategy for high dimensional data sets, which consists of three main components:

- *Initialization partition.*
- *Interrelated iteration.*
  - *Representative pattern discovery.*
  - *Feature pruning.*
- *Class validation.*

Since the volume of features is large and no information regarding the actual partition of the target objects assumed to be available, we cannot directly identify the patterns or significant features. Rather, these goals must be gradually approached. So we first cluster both the target objects and features into several exclusive smaller groups and divide the original matrix into a series of exclusive sub-matrices based on this partition. The goals of pattern discovery and feature identification will be achieved by analyzing the correlations among these sub-matrices.

We analyze the subsets of both the target objects and features and use the relationships thus discovered to post a partial or approximate pattern called a *representative pattern*. We then use this *representative pattern* to direct the elimination of irrelevant features. In turn, the remaining meaningful features will guide further target pattern detection. Thus, we can formulate the problem of pattern discovery in the original data var an interplay between representative target pattern detection and irrelevant feature pruning. Because of the complexity of the matrix, this procedure usually requires several iterations to achieve satisfactory results. The iterative strategy in which these two correlated processes is at the core of this approach to high dimensional data analysis.

The criteria for terminating the series of iterations is determined by evaluating the quality of the data partition. This is achieved in the "class validation" phase by assigning certain statistical measures to the selected features and the related data partition. When a stable and significant pattern of data emerges, the iteration stops, and the selected features and the related data partition become the final result of the process.

3. ITERATIVE STRATEGY FOR PATTERN DISCOVERY

In this section, we will present details relating to the implementation of the iterative pattern-discovery strategy.

3.1 Initialization partition

Let $\widehat{\mathbf{M}} = \{m_{i,j} | i = 1 \sim s, j = 1 \sim t\}$ represent the original matrix, where there are t columns, one for each target object, and s rows, one for each feature, $T = \{1, 2, ..., t\}$ represents the indices of the target objects, and $S = \{1, 2, ..., s\}$ represents the indices of the features. The first step is to cluster features into n_s groups and cluster target objects into n_t groups. We use their indices to denote the groups as $S_x (x = 1, 2, ..., n_s)$ and $T_y (y = 1, 2, ..., n_t)$, which satisfy $\bigcup S_x = S$ while $S_x \bigcap S_{x'} = \emptyset$ $(x \neq x')$ and $\bigcup T_y = T$ while $T_y \bigcap T_{y'} = \emptyset$ $(y \neq y')$. Combining the intersection of target and feature groups above, we have $n_s \times n_t$ exclusive sub-matrices. Each sub-matrix consists of the intersection of one group of target objects and one group of features, denoted as $M_{x,y} = \{m_{i,j} |\ i \in S_x, j \in T_y\}$. Thus $\bigcup M_{x,y} = \widehat{\mathbf{M}}$ and $M_{x,y} \bigcap M_{x',y'} = \emptyset$ $(x \neq x'$ *or* $y \neq y')$.

Several additional aspects of this clustering procedure are addressed below.

3.1.1 Data Standardization

Data sometimes needs to be transformed before being used [18]. For example, features may be measured using different scales, such as centimeters and kilograms. In instances where the range of values differs widely from feature to feature, these differing feature scales can dominate the results of the data analysis. It is therefore common to standardize the data so that all features are on the same scale.

The common approach for data standardization is:

$$m'_{i,j} = \frac{m_{i,j} - \overline{m}_i}{\sigma_i}\ , \tag{1}$$

where

$$\overline{m}_i = \frac{\sum_{j=1}^{t} m_{i,j}}{t}\ ,\ \sigma_i = \frac{\sqrt{\sum_{j=1}^{t} (m_{i,j} - \overline{m}_i)^2}}{t-1}$$

and $m'_{i,j}$ denotes the standardized value for feature i of target object j, $m_{i,j}$ represents the original value for feature i of target object j, t is the number of target objects, $\overline{m}_i$ is the mean of the values for feature i over all target objects, and σ_i is the standard deviation of the i^{th} feature.

3.1.2 Similarity measure

Many methods of cluster analysis depend on some measure of similarity (or distance) between the vectors to be clustered. Although *Euclidean distance* is a popular distance measure for spatial data, the *correlation coefficient* [8] is widely believed to be more suitable for pattern-discovery approaches because it measures the strength of the linear relationship between two vectors. This measure has the advantage of calculating similarity on the basis of the pattern but not the absolute magnitude of the spatial vector. The formula for the correlation coefficient between two vectors $X = (x_1, x_2, ...x_k)$ and $Y = (y_1, y_2, ...y_k)$ is:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{k} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{k} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{k} (y_i - \overline{y})^2}}, \tag{2}$$

where k is the length of vectors X and Y, and

$$\overline{x} = \frac{1}{k} \sum_{i=1}^{k} x_i, \quad \overline{y} = \frac{1}{k} \sum_{i=1}^{k} y_i.$$

We will use the *correlation coefficient* as the similarity measure for the proposed clustering approach so that pattern similarities between features or target objects in each group will be revealed regardless of their spatial proximity.

3.1.3 Estimating the number of clusters

Clustering algorithms, such as K-means and SOM, require users to prescribe the number of clusters for a data set. Although other clustering methods, such as HC, do not need a cluster number as an input parameter, the number of clusters is still indirectly influenced by a user-assigned input parameter, e.g., the affinity threshold. Therefore, the number of clusters of target objects (n_t) and features (n_s) needs to be decided prior to partition.

Some statistical approaches, such as the *Bayesian Model Selection* [13] and the *Gap Statistic* [32] have offered statistical guidelines regarding the number of clusters. However, no definitive method has yet been developed to determine the optimal number of groups and group size. The appropriate ranges of cluster numbers for features and target objects depend on the specific application. For example, in biological applications, experience shows that genes associated with similar functions always involve between several dozen and several hundred entities. Thus, cluster numbers for genes should be selected so that a majority of groups will contain several dozen to several hundred genes. With samples, where a typical matrix contains fewer than 100 samples and a small sample volume may not be representative, the sample cluster number should be below a threshold which generates sample groups of at least five samples. We usually vary the number of sample clusters from 2 to $t/5$ to maintain a good compromise between the number of clusters and the separation among them.

3.2 Pattern discovery and feature pruning

3.2.1 Representative pattern discovery

The ultimate goal of data analysis is to detect a target pattern and to select related features. However, each iteration also has an interim goal, the discovery of a partial or approximate target pattern, called the *representative pattern* of the target object. The representative pattern is selected from elements among the sub-matrices from the preliminary steps according to the following criteria:

- A target group which has a higher correlation measure over all feature groups is chosen as the representation of one target type;

- A contrary set of target groups which mutually have a larger dissimilarity measure is chosen to represent the pattern.

Each sub-matrix $M_{x,y}$ represents the distribution of a subset of target data over a subset of features. If the data in the sub-matrix vary little, then the features involved generally correlate to manifest a single function of the target data. Thus, this set of highly-correlated target data may represent a single type for the entire target pattern, and the features may be good candidates for representing the target types of empirical interest. On the other hand, if the data in a sub-matrix vary much, target objects and features within this sub-matrix will have little correlation and may be considered as noise.

So we measure the coherence of target data and features in a sub-matrix by its *row variance*:

$$Var(M_{x,y}) = \frac{1}{|S_x| \cdot |T_y|} \sum_{i \in S_x} \sum_{j \in T_y} (m_{i,j} - \overline{m}_{i,T_y})^2, \qquad (3)$$

where $\overline{m}_{i,T_y} = \frac{1}{|T_y|} \sum_{j \in T_y} m_{i,j}$. $Var(M_{x,y})$ measures the accumulation of variance for each row in the sub-matrix. Low $Var(M_{x,y})$ value shows the involved subset of target objects all have stable features. If this situation occurs with most groups of features, target group T_y is likely to be chosen as the representative group. We then measure the likelihood of the selection of a given target group as representative group by accumulating its reciprocal of variance over all feature groups. This combined measure is the *representation degree* for each target group, defined as:

$$\gamma_y = \lg |T_y| \sum_{S_x \in S} \frac{\lg |S_x|}{Var(M_{x,y})}. \qquad (4)$$

The target pattern is usually represented by multiple groups of data. The *representation degree* is the measure for single target group only, and objects within a group with a high *representation degree* are more likely to manifest a uniform target type. In order to recognize the full representative pattern, at least one target group should be selected which displays each type. Those target groups which have high *representation degrees* and large mutual dissimilarity between each other should be chosen to form a *representative pattern*. Our method is to first choose the target group which has the highest *representation degree*, denoted by $T_{\hat{y}}$, and then rank other target groups by a score combining their *representation degree* and *dissimilarity* to $T_{\hat{y}}$, as follows:

$$\delta_y = \gamma_y \sum_{S_x \in S} \frac{D(M_{x,y}, M_{x,\hat{y}})}{Var(M_{x,y})}. \qquad (5)$$

where

$$D(M_{x,y}, M_{x,\hat{y}}) = \frac{\sum_{p_1 \in T_y, p_2 \in T_{\hat{y}}} Dis(\vec{m}_{x,p_1}, \vec{m}_{x,p_2})}{|T_y| \times |T_{\hat{y}}|}, \qquad (6)$$

$\vec{m}_{x,p_1}$ is a target object in sub-matrix $M_{x,y}$, and $\vec{m}_{x,p_2}$ is a target object in sub-matrix $M_{x,\hat{y}}$. $D(M_{x,y}, M_{x,\hat{y}})$ is a *group average distance* defined between each two subsets of target objects within sub-matrices ($M_{x,y}$ vs. $M_{x,\hat{y}}$). It measures the difference distribution of two groups of target objects projected on the same group of features. It can also be viewed as defining the various functions of the features on different target objects.

A few top-ranked target groups with high δ scores, along with the group $T_{\hat{y}}$, are selected to form the representative pattern. The number of groups which form the pattern will vary according to the number of empirical types of the data.

3.2.2 Feature Pruning

Another interim goal in each pattern-discovery iteration is to filter out those features which are irrelevant or redundant with respect to the target pattern. In this stage, the representative pattern selected from the last step is used to sort all feature groups.

Because the target pattern consists of several groups, we hope to find features which have a high degree of coherence for all target groups and which display large dissimilarities between each pair of target groups with regard to manifesting the target pattern distribution. All feature groups are therefore ordered in terms of relevance to the representative pattern. We define a ranking criterion, termed *reference quality* (denoted by λ_x), for all feature groups. Assume that the representative pattern includes K groups: $\check{J} = \{J_1, J_2, ...J_K\}$. The formula for λ_x is:

$$\lambda_x = \sum_{J_{y_1}, J_{y_2}} \big[D(M_{x,y_1}, M_{x,y_2}) \times \times \exp\big(-(Var(M_{x,y_1}) + Var(M_{x,y_2}))\big)\big], \qquad (7)$$

where $J_{y_1}, J_{y_2} \in \check{J}$ *and* $J_{y_1} \neq J_{y_2}$. Formula $\exp(-(Var(M_{x,y_1}) + Var(M_{x,y_2})))$ guarantees that λ_x is high when variance values for target groups J_{y_1} and J_{y_2} are both low.

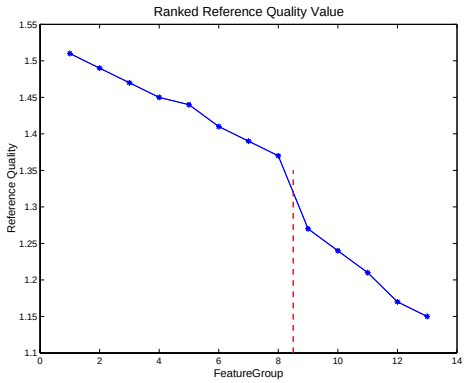

Figure 1: Distribution of ranked reference quality value for all feature groups.

When all feature groups are ranked by this *reference quality*, feature-pruning is performed by eliminating some groups which fall at the end of this ranked list. We examine the groups in the second half of the ranked list, choose a "pruning point" between two groups with the largest difference in *reference quality* values, and remove the feature groups below the pruning point. Figure 1 provides an example in which the line shows the ranked *reference quality* value of all feature groups. As we can see, in the second half of the list (feature groups $7 \sim 13$), the largest difference occurs between groups 8 and 9. Thus, the pruning point will be set between these groups and groups 9 to 13 will be filtered out. The semantic meaning of this pruning criterion is that, while each group between 2 to 8 shows slightly less relevance to the target partition than the previous group, group 9 and following are much less relevant and they can be pruned. It is appropriate to select the pruning point from the second half of the ranked lists so that not too many features will be removed in a single step, particularly when the largest difference appears between the first few groups. Since the representative pattern determined earlier in the process may not exactly match the actual partition, reducing too many features based on the supposed pattern may lead to a biased result.

3.3 Class validation and termination

After one iteration involving detection of representative target pattern and selection of features, a certain number of features will be pruned. The remaining features and the entire set of target objects then form a new matrix and a

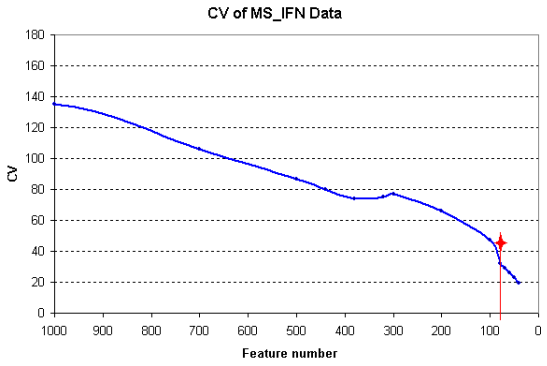

Figure 2: The CV value after each iteration.

new iteration starts.

We will now discuss the issue of determining when sufficient iterations have been performed. Ideally, iterations will be terminated when a stable and significant pattern of target objects has emerged. Thus, the iteration termination criterion involves determining the measurement and threshold which identifies a "stable and significant" pattern.

The purpose of pattern discovery for target objects is based on identifying groups of empirical interesting distributions and patterns in the underlying data. In general, we hope that the objects in a given group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group, and the greater the difference between groups, the better or more distinct the pattern.

As described above, after each iteration, we use the remaining features to classify target objects and then use the coefficient of variation (CV) to measure how "internally-similar and well-separated" this partition is:

$$CV = \frac{1}{N}\sum_{k=1}^{N}\frac{\sigma_k}{||\vec{\mu}_k||}, \tag{8}$$

where N represents the cluster number, μ_k indicates the center of group k, and σ_k represents the standard deviation of group k. Assuming there are t objects ($\vec{v}_1$, $\vec{v}_1$,..., $\vec{v}_t$) in group k, the standard deviation of group k is defined as:

$$\sigma_k = \frac{\sqrt{\sum_{i=1}^{t}||\vec{v}_i - \vec{\mu}_k||^2}}{t-1}.$$

It is clear that, if the dataset contains an "internally-similar and well-separated" partition, the standard deviation of each group will be low, and the CV value is expected to be small. Thus, based on the coefficient of variation, we may conclude that small values of the index indicate the presence of a "good" pattern.

Using the *coefficient of variation* value to identify an iteration termination point is similar to using the *reference quality* to prune features. We examine the values after each iteration and terminate the algorithm after an iteration with a CV value much smaller than the previous. In the example in Figure 2, we have applied this approach to a data set by monitoring the CV until the feature number is very small. This illustrates the change in CV value in relation to feature numbers. In this figure, the CV value drops abruptly after a certain iteration (here, when the feature number is less than 100), and the iteration can then stop. This termination point is indicated by a red star.

Another applicable termination condition involves checking whether the number of features is small enough to guide target-object class prediction. This number highly depend on the type of data. For example, in a typical biological system, the number of genes needed to fully characterize a macroscopic phenotype and the factors determining this number are often unclear. Experiments also show that, for certain data, gene numbers varying from $10 \sim 200$ can all serve as good predictors [16]. For our microarray data experiments, we have chosen 100 as a compromise termination number; e.g. when the number of genes falls below 100, the iteration stops. This termination condition is used only as a supplementary criterion.

Features that remain will be regarded as the selected features resulting from this pattern-discovery approach. They are then used to classify the target objects for a final result. Since the number of features is relatively small, the traditional clustering methods can be applied to the selected features. The remaining features can also be treated as "predictors" to establish cluster labels for the next batch of target objects.

4. PERFORMANCE EVALUATION

In this section, we will analyze the effectiveness of the proposed approach through experiments on various data sets. Different measurement techniques are tested and illustrated.

4.1 External evaluation criteria

4.1.1 Rand index

A measurement of "agreement" (called the "Rand Index" [26, 36]) between the ground-truth of the target partition and the pattern discovery result was used to evaluate the performance of the algorithm. Given a set of target objects $T = \{t_1, t_2, ..., t_{n_t}\}$, suppose $P = \{p_1, p_2, ..., p_k\}$ is the actual partition according to the target types of empirical interest, and $Q = \{q_1, q_2, ..., q_m\}$ is a partition of target objects resulting from the clustering algorithm which satisfies $T = \bigcup_{i=1}^{k} p_i = \bigcup_{j=1}^{m} q_j$, $p_i \bigcap p_{i'} = \emptyset$ $(1 \le i, i' \le k)$ *and* $q_j \bigcap q_{j'} = \emptyset$ $(1 \le j, j' \le m)$. Let $\boldsymbol{a}$ represent the number of pairs of objects that are in the same class in P and in the same cluster in Q, $\boldsymbol{b}$ represent the number of pairs of objects that are in the same class in P but not in the same cluster in Q, $\boldsymbol{c}$ be the number of pairs of objects that are in the same class in Q but not in the same cluster in P, and $\boldsymbol{d}$ be the number of pairs of objects that are in different classes in P and in different clusters in Q. Thus, $\boldsymbol{a}$ and $\boldsymbol{d}$ measure the agreement of two partitions, while $\boldsymbol{b}$ and $\boldsymbol{c}$ indicate disagreement. The formula of the *Rand Index* [26] is:

$$RI = \frac{a+d}{a+b+c+d}. \tag{9}$$

The *Rand Index* lies between 0 and 1. When the two partitions match perfectly, the *Rand Index* is 1. In our experiments, we calculate a *Rand Index* value between the ground-truth and the result of each potential method to evaluate the quality of the clustering algorithms. In these tests, a higher the *Rand Index* value indicates better algorithm performs.

4.1.2 Interactive visualization

A linear mapping tool [38] which maps the n-dimensional dataset onto two-dimensional space is used to view the changes in target-data distribution during the iterative process.

Let vector $\vec{P_g^*} = (x_{g1}, x_{g2}, \ldots, x_{gn})$ represent a data element in the n-dimensional space. Equation (10) describes the mapping of $\vec{P_g^*}$ onto a two-dimensional point $\vec{Q_g^*}$:

$$\vec{Q_g^*} = \sum_{i=1}^{n} (\lambda_i * x_{gi}) \vec{S_i} \qquad \lambda_i \in [-1, 1] \qquad (10)$$

where n is the number of dimensions of the input space, and $\vec{S_i}$ $(i = 1, 2, \ldots, n)$ are unit vectors which divide the center circle of the display into n equal directions, i.e., $\vec{S_i} = 2\pi/n{*}i$. The mapping Formula (10) replicates the correlation relationship of the input space onto the two-dimensional images. Note that point $(0, 0, \ldots, 0)$ in the input space will be mapped onto the two-dimensional center $(0, 0)$ (assuming all dimension weights are equal). Additionally, all points in the format $(a, a, \ldots, a)$ will also be mapped to the center. If $\vec{X}$ and $\vec{Y}$ have the same pattern; i.e., ratios of each mapped pair, these vectors will be mapped onto a straight line across the center of the $2D$ display space. All vectors with same pattern as $\vec{X}$ and $\vec{Y}$ will be mapped onto that line. This mapping method takes the advantage of graphical visualization techniques to reveal the underlying data patterns.

4.2 Experimental Results

We will now present experimental results using three microarray data sets. The first two data sets are from a study of multiple-sclerosis patients collected by the Neurology and Pharmaceutical Sciences Departments of the State University of New York at Buffalo [23]. Multiple sclerosis (MS) is a chronic, relapsing, inflammatory disease, and interferon-β (IFN-β) has offered the main treatment for MS over the last decade [37]. The MS dataset includes two groups: the MS_IFN group, containing 28 samples (14 MS, 14 IFN), and the MS_CON group, containing 30 samples (15 MS, 15 Control). Each sample is measured over 4132 genes. The third data set is based on a collection of leukemia patient samples reported in (Golub et al., 1999) [16]. The matrix includes 72 samples (47 ALL vs. 25 AML). Each sample is measured over 7129 genes. The ground-truth of the partition, which includes such information as how many samples belong to each cluster and the cluster label for each sample, is used only to evaluate the experimental results.

To evaluate the performance of the proposed algorithm, we compared its performance in classifying the samples with several popular microarray data analysis tools such as CLUSFAVOR [25], J-Express [27], CIT [10] and CLUTO [29]. The major unsupervised data analysis methods involved include: K-means, self-organizing maps (SOM), clustering algorithms based on the graph-partitioning paradigm (CLUTO) and the dimensionality reduction method PCA (principal component analysis).

Table 1 provides pattern matching result obtained by directly applying the above algorithms to high gene-dimension data without an iterative process. All these algorithms were applied to the matrix after data standardization according to Equation (1). This table indicates that performance of SOM is slightly superior for the two MS datasets. The CLUTO performed better with the leukemia datasets. However, none of these methods resulted in a very good matching rate. Since the central idea of principal component analysis (PCA) is to reduce the dimensionality of the data set while retaining as much as possible the variation in the data set, principal components (PC's) do not necessarily capture the cluster structure of the data [36]. So the results of clustering with PCA did not improve the results significantly.

Data Set	MS_IFN	MS_CON	Leukemia
Sample #	28	30	72
k-means	0.4815	0.4851	0.5070
SOM	0.4815	0.4920	0.5027
k-means with PCA	0.4841	0.4851	0.5246
SOM with PCA	0.5238	0.5402	0.5180
CLUTO	0.4815	0.4828	0.5121

Table 1: Rand Index value reached by applying several microarray data analysis tools.

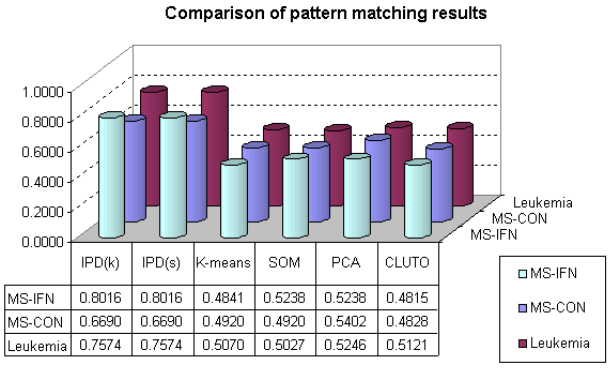

Figure 3: Rand Index values for the Iterative Pattern-Discovery approach and comparison with other methods. IPD(k) stands for Iterative Pattern-Discovery with K-means as basic clustering algorithm. IPD(s) stands for Iterative Pattern-Discovery with SOM as basic clustering algorithm.

The proposed iterative pattern-discovery approach was also applied to the same gene expression matrices. The results obtained depend on the basic clustering algorithm in the preliminary steps. Figure 3 provides clustering results of the multiple sclerosis and leukemia datasets. The first two column are the *Rand Index* values achieved by iterative pattern-discovery approach. These results indicate that, the index is consistently higher than the results obtained by directly applying others methods.

In Figure 4, the interactive visualization tool is used to show the distribution of samples during the iterative pattern-discovery procedure. As indicated by this figure, prior to the application of the iterative approach, the samples are uniformly scattered, with no obvious clusters. As the iterations proceed, sample clusters progressively emerge until in Figure 4(D), the samples are clearly separated into two groups. The green and red dots indicate the actual partition of the samples, while the two dashed circles show the clusters resulting from the iterative approach, with arrows pointing out the incorrectly-classified samples. This visualization provides a clear illustration of the iterative process.

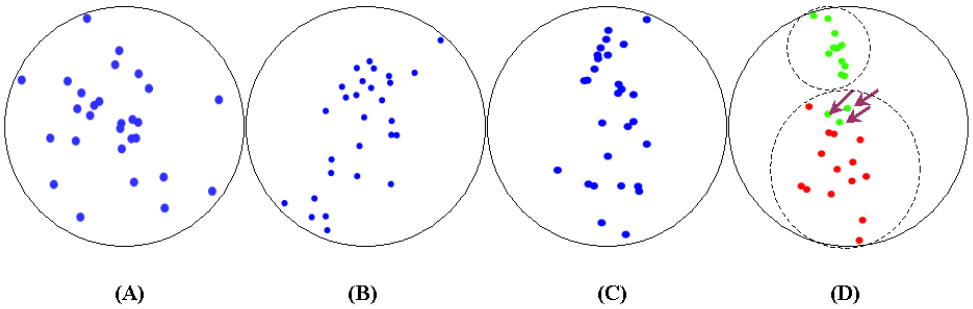

Figure 4: The iterative approach as applied to the MS_IFN group. (A) shows the distribution of the original 28 samples. Each point represents a sample mapped from the intensity vectors of 4132 genes. (B) shows the distribution of the 28 samples over 2015 genes. (C) shows the distribution of the 28 samples over 312 genes. (D) shows the distribution of the same 28 samples after the iterative approach. The 4132 genes have been reduced to 96 genes, therefore each sample is a 96-dimension vector.

Here, it selected 96 genes as features and classified 28 samples into two groups. 11 samples are in group one, matching the MS disease samples. Another 17 samples are in group two, of these, 14 are from the IFN treatment group and 3 are incorrectly matched.

Figures 3 and 4 therefore illustrate the effectiveness of the iterative pattern-discovery method for such high-dimensional gene data.

5. CONCLUSION

In this paper, we have presented an iterative strategy for pattern discovery in high dimensional data sets, motivated by the needs of emerging microarray data analysis. The strategy is designed to improve the unsupervised clustering or classification performance for various kinds of data which have the following properties:

- The number of target data is limited but the feature dimension is very large.
- Large volumes of irrelevant and redundant features prevent accurate grouping of target data;
- Analyzing over one dimension object can enhance detecting meaningful patterns of another dimension.

This approach can detect significant patterns within target data sets while dynamically pruning out irrelevant features and selecting significant features which manifest the patterns of actual empirical interest. We have demonstrated the effectiveness of this approach through experiments conducted with two multiple-sclerosis data sets and a leukemia data set. These experiments indicate that this appears to be a promising approach for unsupervised pattern discovery on high-dimensional data sets.

6. REFERENCES

[1] Aggarwal, Charu C., Wolf, Joel L., Yu, Philip S., Procopiuc, Cecilia and Park, Jong Soo. Fast algorithms for projected clustering. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 61–72, Philadephia, Pennsylvania, USA, 1999.

[2] Agrawal, Rakesh, Gehrke, Johannes, Gunopulos, Dimitrios and Raghavan, Prabhakar. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[3] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*, Vol. 96(12):6745–6750, June 1999.

[4] Ben-Dor A., Friedman N. and Yakhini Z. Class discovery in gene expression data. In *Proc. Fifth Annual Inter. Conf. on Computational Molecular Biology (RECOMB 2001)*, pages 31–38. ACM Press, 2001.

[5] Brown M.P.S., Grundy W.N., Lin D., Cristianini N., Sugnet C.W., Furey T.S., Ares M.Jr. and Haussler D. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl. Acad. Sci.*, 97(1):262–267, January 2000.

[6] Chakrabarti, Kaushik and Mehrotra, Sharad. Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces. In *The VLDB Journal*, pages 89–100, 2000.

[7] DeRisi J. *et al.* Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14:457–460, 1996.

[8] Devore, Jay L. *Probability and Statistics for Engineering and Sciences*. Brook/Cole Publishing Company, 1991.

[9] Ding, Chris. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proc. of International Conference on Computational Molecular Biology (RECOMB)*, pages 127–136, Washington, DC., April 2002.

[10] Dysvik B. and Jonassen I. J-Express: exploring gene expression data using Java. *Bioinfomatics*, 17(4):369–370, 2001. Applications Note.

[11] Eisen M.B., Spellman P.T., Brown P.O. and Botstein D. Cluster analysis and display of genome-wide

expression patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, 1998.

[12] Ermolaeva O., Rastogi M. *et al.* Data management and analysis for gene expression arrays. *Nature Genetics*, 20:19–23, 1998.

[13] Fraley C. and Raftery A.E. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, 1998.

[14] Furey T.S., Cristianini N., Duffy N., Bednarski D.W., Schummer M., and Haussler D. Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. *Bioinformatics*, Vol.16(10):909–914, 2000.

[15] Getz G., Levine E. and Domany E. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, Vol. 97(22):12079–12084, October 2000.

[16] Golub T.R., Slonim D.K. *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.

[17] Groth, Robert. *Data Mining: Building Competitive Advantage*. Prentice Hall PTR, 1999.

[18] Han, Jiawei and Kamber, Micheline. *Data Mining: Concept and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, August 2000.

[19] Jiang S., Tang C., Zhang L., Zhang A. and Ramanathan M. A maximum entropy approach to classifying gene array data sets. In *Proc. of Workshop on Data mining for genomics, First SIAM International Conference on Data Mining*, 2001.

[20] Kohonen T. *Self-Organization and Associative Memory*. Spring-Verlag, Berlin, 1984.

[21] Li, Wentian. Zipf's Law in Importance of Genes for Cancer Classification Using Microarray Data. Lab of Statistical Genetics, Rockefeller University, April 2001.

[22] MacQueen J.B. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Univ.of California, Berkeley, 1967. Univ.of California Press, Berkeley.

[23] Nguyen LT., Ramanathan M., Munschauer F., Brownscheidle C., Krantz S., Umhauer M., et al. Flow cytometric analysis of in vitro proinflammatory cytokine secretion in peripheral blood from multiple sclerosis patients. *J Clin Immunol*, 19(3):179–185, 1999.

[24] Park P.J., Pagano M., and Bonetti M. A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. In *Pacific Symposium on Biocomputing*, pages 52–63, 2001.

[25] Peterson Leif E. Factor analysis of cluster-specific gene expression levels from cdna microarrays. *Computer Methods and Programs in Biomedicine (in press)*, 2002.

[26] Rand, W.M. Objective criteria for evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

[27] Rhodes, D.R., Miller, J.C., Haab, B.B., Furge, K.A. CIT: Identification of Differentially Expressed Clusters of Genes from Microarray Data. *Bioinformatics*, 18:205–206, 2001.

[28] Schena M., Shalon D., Davis R.W. and Brown P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[29] Schloegel, Kirk, Karypis, George. *CRPC Parallel Computing Handbook*, chapter Graph Partitioning For High Performance Scientific Simulations. Morgan Kaufmann, 2000.

[30] Tamayo P., Solni D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S. and Golub T.R. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, Vol. 96(6):2907–2912, March 1999.

[31] Tavazoie, S., Hughes, D., Campbell, M.J., Cho, R.J. and Church, G.M. Systematic determination of genetic network architecture. *Nature Genet*, pages 281–285, 1999.

[32] Tibshirani R., Walther G. and Hastie T. Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Dept of Statistics, Stanford Univ., 2000.

[33] Tusher V.G., Tibshirani R. and Chu G. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proc. Natl. Acad. Sci. USA*, Vol. 98(9):5116–5121, April 2001.

[34] Xing E.P. and Karp R.M. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, Vol. 17(1):306–315, 2001.

[35] Yang, Jiong, Wang, Wei, Wang, Haixun and Yu, Philip S. delta-cluster: Capturing Subspace Correlation in a Large Data Set. In *Proceedings of 18th International Conference on Data Engineering (ICDE 2002)*, pages 517–528, 2002.

[36] Yeung, Ka Yee and Ruzzo, Walter L. An empirical study on principal component analysis for clustering gene expression data. Technical Report UW-CSE-2000-11-03, Department of Computer Science & Engineering, University of Washington, 2000.

[37] Yong V., Chabot S., Stuve Q. and Williams G. Interferon beta in the treatment of multiple sclerosis: mechanisms of action. *Neurology*, 51:682–689, 1998.

[38] Zhang L., Tang C., Shi Y., Song Y., Zhang A. and Ramanathan M. VizCluster: An Interactive Visualization Approach to Cluster Analysis and Its Application on Microarray Data. In *Second SIAM International Conference on Data Mining (SDM'2002)*, pages 19–40, Arlington, Virginia, April 11-13 2002.