Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

PCGEN: A Practical Approach to Projected Clustering and its Application to Gene Expression Data

Mohamed Bouguessa and Shengrui Wang
Department of Computer Science, University of Sherbrooke
Quebec, Canada, J1K 2R1
{mohamed.bouguessa, shengrui.wang}@usherbrooke.ca

Abstract—**Clustering samples in gene expression data has always been a major challenge because of the high dimensionality of the input space (typically in the tens of thousands) and the small number of samples (typically less than a hundred). Moreover, clusters may hide in subspaces with very low dimensionalities. Most existing clustering algorithms become substantially inefficient if the required similarity measure is computed between data points in the full-dimensional space. These challenges motivate our effort to propose a new and efficient partitional distance-based projected clustering algorithm for clustering samples in gene expression data. Our algorithm is capable of detecting projected clusters of extremely low dimensionality embedded in a high-dimensional space and avoids the computation of the distance in the full-dimensional space. The suitability of our proposal has been demonstrated through an empirical study using public microarray datasets.**

I. Introduction

Data mining is the process of extracting potentially useful information from a dataset [1]. Clustering is a popular data mining technique which is intended to help the user discover and understand the structure or grouping of the data in the set according to a certain similarity measure [2]. Clustering algorithms usually employ a distance metric (e.g., Euclidean) or a similarity measure in order to partition the database so that the data points in each partition are more similar than points in different partitions.

The commonly used Euclidean distance, while computationally simple, requires similar objects to have close values in all dimensions. However, with the high-dimensional data commonly encountered nowadays, the concept of similarity between objects in the full-dimensional space is often invalid and generally not helpful. Recent theoretical results [3] reveal that in high-dimensional data, the distance between any two data points becomes almost the same, making it difficult to differentiate similar data points from dissimilar ones.

Feature selection techniques are commonly utilized as a preprocessing stage for clustering in order to overcome the curse of dimensionality. The most informative dimensions are selected by eliminating irrelevant and redundant one. Such techniques speed up clustering algorithms and improve their performance [4]. Nevertheless, in real-life applications, different clusters may exist in different subspaces spanned by different dimensions. In such cases, dimension reduction using a conventional feature selection technique may lead to substantial information loss [5]. Consequently, it can generate clusters that may not reflect well the original clusters properties [6]. A prominent example is the application of cluster analysis to gene expression data.

Genome expression data reflects the level of activity of several genes in parallel under different biochemical conditions [7]. Usually, the format of such a dataset conforms to the normal data format of machine learning and data mining, where a gene can be regarded as a feature or attribute and a sample (e.g., different experiments, test subjects, etc.) as an object or data point [7]. The challenge in dealing with gene expression data lies in the fact that there are order of magnitude differences between the number of samples (typically less than a hundred) and the number of genes (typically tens of thousands) that are studied [8]. In addition, it is meaningful to cluster either genes or samples, which is a particular characteristic of gene expression data analysis [8]. In this paper we will focus exclusively on sample clustering

The aim of sample clustering is to cluster samples into homogenous groups that may correspond to particular macroscopic phenotypes, such as clinical syndromes or cancer types [9]. It is more difficult than gene clustering in a sense, because of the curse of dimensionality (small sample volume and high feature dimensionality). However, sample clustering can be very valuable in clinical and mechanistic studies. For example, in cancer diagnosis, the samples may represent test subjects. The ultimate goal of the clustering is then to distinguish between healthy and ill patients [9]. On the other hand, due to the extreme sparsity of the data, biologists are faced with the problem of choosing the smallest number of genes which potentially contain biologically relevant and meaningful information.

Current research in molecular biology holds that only a subset of genes participates in any cellular process of interest and that a cellular process takes place only in a subset of the samples. Consequently, most of the genes collected may not necessarily be of interest. Only a small percentage (less than 5 percent) of them manifest meaningful sample structures [9]. For instance, two genes have similar expression patterns only in a subset of samples where certain regulating factors are present. In the other samples, the two genes may express differently [10]. In other words, clusters of samples may hide in certain subspaces of genes. This requires a projected

1-4244-0705-2/07/$20.00 ©2007 IEEE

clustering technique capable of capturing clusters formed by a subset of samples across a subset of genes.

Projected clustering exploits the fact that different groups of data points are correlated along different sets of dimensions in high dimensional datasets. The clusters produced by such algorithms are called "projected clusters". A projected cluster is a subset *P* of data points, together with a subspace of dimensions[1] *D*, such that the points in *P* are closely clustered in *D* [5]. The dimensions in which a cluster exists are called relevant dimensions while the others are called irrelevant ones.

For the purpose of illustration, we have generated a dataset composed of *N* data points in 10-dimensional space, as shown in Figure 1. The dataset contains four projected clusters, each having their own relevant dimensions (e.g., cluster 1 is represented by the couple {*P1, D1*} = {(*x1, ..., xa*), (A_3, A_5, A_6, A_8, A_{10})}). For each relevant dimension of a cluster, points in the cluster are distributed according to a normal distribution, while in the remaining dimensions, the points are distributed sparsely. In addition, there are two irrelevant dimensions A_4 and A_7 in which all the data points are sparsely distributed, i.e. no cluster structure exist.

For such an example, projected clustering methods are able to capture clusters spanned in different subspaces while traditional clustering algorithms fail to do so. Although feature selection techniques can reduce the dimensionality of the data by eliminating irrelevant attributes such as A_4 and A_7, there is an enormous risk that they will also eliminate relevant attributes such as A_1. This is due to the presence of many sparse data points in A_1, where cluster 2 is in fact present.

The remainder of this paper is organized as follows. In section 2, we provide a brief overview of recent projected clustering algorithms. Section 3 describes our projected clustering algorithm in detail. Section 4 presents the experiments and the performance results on real datasets. Our conclusion is given in Section 5.

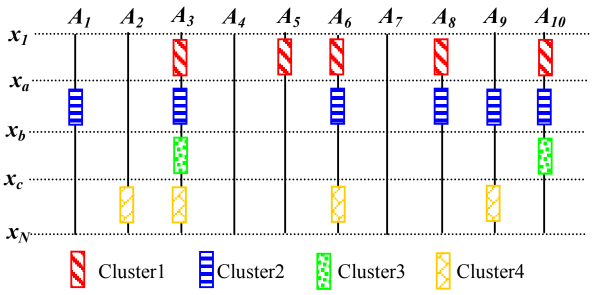

Fig. 1. Example of projected clusters in high-dimensional data.

II. Related Work

The problem of finding projected clusters has been addressed in [5]. The partitional algorithm PROCLUS, which is a variant of the k-medoid method, iteratively computes a good medoid for each cluster. With the set of medoids, PROCLUS finds the subspace dimensions for each cluster by examining the locality of the space near it. After the subspace has been determined, each data point is assigned to the cluster of the nearest medoid. The algorithm is run until the sum of intracluster distances ceases to change. ORCLUS [11] is an extended version of PROCLUS that looks for non-axis parallel subspaces. For this purpose, ORCLUS makes use of Singular Value Decomposition (SVD) to transform the data to a new coordinate system and select principal components. PROCLUS and ORCLUS successfully introduce a methodology for looking at different subspaces for different clusters and continue to inspire novel approaches.

A limitation of these two approaches is that the process of forming the locality is based on the full dimensionality of the space. However, due to the huge number of attributes (genes) in gene expression data, it makes no sense to look for neighbors in high-dimensional space [3]. In addition, PROCLUS and ORCLUS require the user to provide the average dimensionality of the subspace, which also is very difficult to do in gene expression data.

HARP [12] is a hierarchical projected-clustering algorithm based on the assumption that if two data points are similar in high-dimensional space, they have a high probability of belonging to the same cluster in lower-dimensional space. However, it has been shown in [3] that as dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point. This is the case when the subspace clusters have few relevant dimensions, and the accuracy of HARP thus deteriorates severely in this situation. This effect on HARP's performance was also observed by Yip et al. in [13]. In addition, the running time of HARP can be expected to be very long due to its hierarchical nature [13]. On the other hand, HARP has the interesting property of avoiding the use of input parameters, whose values are difficult to set.

A density-based algorithm named EPC is proposed in [14] for projected clustering. EPC performs projected clustering by histogram construction. By iteratively lowering a threshold, dense regions are identified in each histogram. A "signature" is generated for each data point corresponding to some region in some subspace. Projected clusters are uncovered by identifying signatures with a large number of data points [14]. While EPC avoids the computation of distance between data points in the full-dimensional space, it suffers from the curse of dimensionality. In our experiments, we have observed that when the dimensionality of the data space increases and the number of relevant dimensions of clusters decreases, the accuracy of EPC is greatly affected. In addition, when we perform sample clustering in gene

[1] The terms "dimension" and "attribute" refer to the same concept and will be used interchangeably in this paper.

expression data, it is difficult to detect dense regions based on histograms because the number of data points (samples) is too small to build a histogram that could faithfully reflect the distribution of attribute values (genes). A thorough survey of the above algorithms and others proposed for the projected clustering problem can be found in [15].

A. Contribution

Clusters in gene expression data contain an extremely low percentage of relevant attributes (less than 5% of all the genes) [9][10][13]. Yip et al. [13] observed that most of the existing projected clustering algorithms are unable to identify clusters with such low dimensionality. In addition, the presence of a large number of irrelevant attributes (genes) and the limited number of data points (samples) often prevent accurate grouping of the samples.

These observations motivate our effort to propose a novel projected clustering algorithm, called PCGEN, for clustering biological samples using gene expression microarray data. Our algorithm is able to detect projected clusters of very low dimensionality embedded in high-dimensional space and avoids the computation of the distance in the full-dimensional space.

III. The Algorithm PCGEN

Let *DB* be a data set of *d*-dimensional points, where the set of attributes is denoted by $A=\{A_1, A_2, ..., A_d\}$. Let $X=\{x_1, x_2, ..., x_N\}$ be the set of *N* data points, where $x_i=(x_{i1}, ..., x_{ij}, ..., x_{id})$. Each x_{ij} $(i=1, ..., N; j=1, ..., d)$ corresponds to the value of data point x_i on attribute A_j. In what follows, we will call x_{ij} a *1-d point*. Suppose that *nc* is the number of clusters in *DB*. A projected cluster C_S $(S = 1, ..., nc)$ containing N_S data points is defined in a d_S-dimensional subspace formed by the set A_S $(A_S \subseteq A)$ of its relevant attributes. The remaining set $A - A_S$ represents the irrelevant attributes of C_S. In order to identify clusters in different subspaces, PCGEN proceeds in two phases. The main focus of the first phase is to detect dense regions in each dimension. Starting from the results of the first phase, the second phase aims to discover clusters in different subspaces. The clustering process is based on the *K*-means algorithm, with the computation of distance restricted to subsets of attributes where object values are dense.

A. Dense Regions Detection

In high-dimensional data, irrelevant attributes contain noise and data points with sparse values, while relevant ones may exhibit some cluster structure. By cluster structure we mean a region that has a higher density of points than its surrounding regions. These dense regions represent the *1-d* projection of some clusters. Our assumption is based on the downward closure property of density, which indicates that if there are dense regions in *k* dimensions, there are dense units in all *(k-1)* dimensional projections [1]. Hence, it is clear that by detecting dense regions in each attribute we are able to discriminate between relevant and irrelevant attributes. Such information will be very useful in phase 2 of PCGEN.

In order to detect densely populated regions in each attribute, we compute a sparseness degree λ_{ij} for each *1-d point* x_{ij} by measuring the variance of its *k* nearest (*1-d point*) neighbors.

Definition 1. The sparseness degree of x_{ij} is defined as

$$\lambda_{ij} = \frac{\sum_{y \in p_i^j(x_{ij})} (y - C_i^j)^2}{k+1}$$

where $p_i^j(x_{ij}) = \{nn_k^j(x_{ij}) \cup x_{ij}\}$

and $|p_i^j(x_{ij})| = k+1$.

$nn_k^j(x_{ij})$ denotes the set of *k*-nn of x_{ij} in attribute A_j and C_i^j is the center of the set $p_i^j(x_{ij})$; i.e.,

$$C_i^j = \frac{\sum_{y \in p_i^j(x_{ij})} y}{k+1}$$

Intuitively, a large value of λ_{ij} means that x_{ij} belongs to a sparse region, while a small one indicates that x_{ij} belongs to a dense region.

Calculation of the *k* nearest neighbors is, in general, an expensive task, especially when the number of data points *N* is very large. However, since we are searching for the *k* nearest neighbors in a 1-dimensional space, we can perform the task in an efficient way by pre-sorting the values in each attribute and limiting the number of distance comparisons to a maximum of *2k* values.

In order to identify dense regions in each attribute, we are interested in all sets of x_{ij} having a small sparseness degree, determined by a pre-defined threshold $\varepsilon \in \Re^+$.

Definition 2. Let $\varepsilon \in \Re^+$, where ε is a density threshold
If $\lambda_{ij} < \varepsilon$ then $z_{ij} = 1$ and x_{ij} belong to a dense region;
else $z_{ij} = 0$ and x_{ij} belong to a sparse region.

From definition 2, we obtain a binary matrix $Z_{(N*d)}$ which contains the information on whether each data point falls into a dense region of an attribute. For example, Figure 2 illustrates the matrix *Z* for the data used in the example in Section 1. The binary weight z_{ij} will play an important role in determining the relevancy of each attribute as well as in estimating the similarity between samples in phase 2 of PCGEN.

Definitions 1 and 2 suit our purpose of detecting dense regions in 1-dimensional space because they are based on the fact that the values of the *1-d* projection of data points onto relevant dimensions will be concentrated in small ranges of values.

It is clear that the computation of the binary weights z_{ij} depends on the two input parameters ε and *k*. Although it is difficult to formulate and obtain optimal values for these parameters, it is easy for us to propose guidelines for their estimation. In practice, the values of the sparseness degree λ_{ij}, the indicator for dense regions, vary significantly

		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
Cluster 1	x_1	0	0	1	0	1	1	0	1	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	0	0	1	0	1	1	0	1	0	1
Cluster 2	x_a	1	0	1	0	0	1	0	1	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	1	0	1	0	0	1	0	1	1	1
Cluster 3	x_b	0	0	1	0	0	0	0	0	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	0	0	1	0	0	0	0	0	1	1
Cluster 4	x_c	0	1	1	0	0	1	0	0	1	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	x_N	0	1	1	0	0	1	0	0	1	0

Fig. 2. The matrix Z(N*d).

depending on the ranges of values in each attribute. In order to have meaningful values of λ_{ij}, we suggest standardizing the values in each attribute with mean 0 and variance 1 before calculating it. According to our experiments, $\lambda_{ij} \leq 0.1$ is a good indication that x_{ij} belongs to a dense region. Therefore, setting $0<\varepsilon \leq 0.1$ is a reasonable choice.

The choice of k can be made in a more straightforward fashion. If k is too small, the sparseness degrees λ_{ij} are not meaningful, since a *1-d* point in a dense region might have a similar sparseness degree value to a *1-d* point in a sparse region. If k is too high, the same phenomenon may occur. Obviously, the parameter k is related to the expected minimum cluster size and should be much smaller than the number of objects N in the data. To gain a clear idea of the sparseness of the neighborhood of a point, we suggest choosing the value of k close to $\sqrt{N}$ as a general guideline. Further investigation into the selection of these parameters is needed.

Once dense regions are detected in each attribute, we turn to the problem of discovering clusters spanning different subspaces.

B. Discovering Projected Clusters

The main focus of Phase 2 of PCGEN is to perform projected clustering and detect relevant dimensions for each cluster. For this purpose we use the *K*-means algorithm and exploit the properties of the matrix $Z_{(N*d)}$. The *K*-means partitions the data into a number of clusters, each of which is represented by a center. A data point is assigned to a cluster using a distance function, e.g., Euclidean distance, to calculate its distance from the center of the cluster. However, this is not an effective approach with high-dimensional data, because each dimension is equally weighted in computing the distance between two points. To solve this problem, we associate the binary weights z_{ij} $(i = 1,..., N;\ j = 1, ..., d)$ in the matrix Z to the Euclidian distance. This makes the distance measure more effective because the computation of distance is restricted to subsets where the object values are dense.

Formally, this weighted Euclidean distance between a point x_i and the cluster center v_c $(c = 1,..., nc)$ is defined as:

$$dist(x_i, v_c) = \sqrt{\sum_{j=1}^{d} z_{ij} \times (x_{ij} - v_{cj})^2} \qquad (1)$$

The use of the matrix Z in the *K*-means algorithm to compute 1) the distance between cluster centers and data points and 2) the centroid coordinates (see steps 3 and 4 in Figure 3) avoids the computation of the distance in the full-dimensional space and clusters the data points in a more efficient way.

Once the data points are clustered, we turn to the problem of detecting relevant dimensions for each cluster. For this purpose, we make use of the density information stored in the matrix Z to determine how well a dimension contributes to the formation of the obtained clusters. In fact, the sum of the binary weights of the data points belonging to the same cluster over each dimension gives us a meaningful measure of the relevance of each dimension to the cluster. Based on this observation, we propose a relevance index W_{Sj} for each dimension in cluster C_S. The index W_{Sj} for the dimension j $(j = 1, ..., d)$ in cluster C_S is defined as follows:

$$W_{Sj} = \frac{\sum_{z_i \in C_S} z_{ij}}{|C_S|} \qquad (2)$$

The value of the index is always between 0 and 1. The index gives a large value (close to 1) when the dimension is relevant to the cluster. On the other hand, an irrelevant dimension receives a very small index value (close to 0).

Definition 3. Let $\delta \in \left]0,1\right]$, a dimension A_j is considered δ relevant for the projected cluster C_S if the following holds:

$$W_{Sj} > \delta$$

In definition 3, δ is a user-defined parameter that controls the degree of relevancy of the dimension A_j to the cluster C_S. It is clear that the more relevant the dimension to the cluster, the larger the value of the relevance index. Since W_{Sj} is a relative measure, it is not difficult to choose an appropriate value for δ. In order to ensure high degree of relevance, setting $\delta \geq 0.8$, in general several, is a practical choice. The PCGEN algorithm is summarized in Figure 3.

As we can see from Figure 3, our clustering process is based on the basic *K*-means algorithm with two significant modifications. The first is the use of the weighted Euclidean distance in step 3; the second, the use of the matrix Z in step 4 for computing the cluster centers. The use of the matrix Z restricts the computation of distance and cluster centers to those subsets of attributes in which the data points belong to dense regions. The efficiency of PCGEN on gene expression data is demonstrated in the following section.

IV. Empirical Evaluation

In this section we experimentally evaluate the suitability of our algorithm by comparing it with HARP and PROCLUS on three gene expression data sets.

```
Input:
Data set DB, number of clusters nc, the parameter k,
and the thresholds $\varepsilon$ and $\delta$ .
Output:
Cluster centers v, the matrix $U_{(N*nc)}$ of the membership
degrees of each data point in each cluster and the sets
$A_S$ of the relevant dimensions of each cluster.

1. Based on Definition 1 and Definition 2, compute
   the matrix Z.
2. Start with initial cluster centers $v_c^0$ $(c = 1, \dots, nc)$
   selected randomly from DB.
3. Compute the membership matrix $U_{(N*nc)}$
     for $i = 1, \dots, N$ and $m = 1, \dots, nc$
     if $dist(x_i, v_c) < dist(x_i, v_m)$ then $u_{im} = 0$
                                 else $u_{im} = 1$
4. Compute the cluster centers
     $v_c^1 = \frac{\sum_{i=1}^{N}(u_{ic} \times z_i \times x_i)}{\sum_{i=1}^{N} u_{ic}}; c = 1, \dots, nc$
5. Iterate 3 and 4 until convergence (i.e., no change
   in centroid coordinates).
6. Based on Definition 3, detect the sets $A_S$ of
   relevant dimensions for each projected cluster $C_S$.
7. End of the algorithm.
```

Fig. 3. The PCGEN Algorithm.

A. Dataset Illustrations

Simple illustrations of the datasets used in this paper for exploring the performance of our algorithm are given in this subsection. All of the datasets were downloaded from: http://sdmc.lit.org.sg/GEDatasets/Datasets.

ALL-AML leukemia: This data set contains 7129 genes and 72 samples from acute leukemia patients. The samples are grouped into two clusters: 47 of the samples are from patients with acute lymphoblastic leukemia (ALL) while 25 are from patients with acute myeloid leukemia (AML).

MLL leukemia: This data set contains three clusters corresponding to different types of leukemia: ALL, AML and mixed lineage leukemia (MLL). The number of genes is 12582 and the number of samples is 72 (24 ALL, 20 MLL and 28 AML).

Lung cancer: The set contains 181 tissue samples, each with 12533 genes. The samples are grouped into two clusters: 31 samples for pleural mesothelioma (MPM) and 150 for adenocarcinoma (ADCA) of the lung.

B. Results

In order to evaluate the quality of the results of PCGEN, HARP and PROCLUS, we used the class labels as ground truth. The Hubert-Arabie Adjusted Rand Index (ARI) [16], which measures the similarity between the generated partition (*GP*) of data points and the real partition (*RP*), is used as performance measure. ARI has been shown to be the most desirable index for measuring agreement between two partitions [17]. A deeper investigation on the properties of ARI can de found in [18]. The Hubert-Arabie Adjusted Rand Index is defined as follows:

$$ARI(RP,GP)=$$

$$\frac{\binom{N}{2}(N_{11}+N_{00})-[(N_{11}+N_{10})(N_{11}+N_{01})+(N_{01}+N_{00})(N_{10}+N_{00})]}{\binom{N}{2}^2-[(N_{11}+N_{10})(N_{11}+N_{01})+(N_{01}+N_{00})(N_{10}+N_{00})]} \qquad (3)$$

where N_{11}, N_{10}, N_{01} and N_{00} are the number of object pairs that are in the same cluster in both *RP* and *GP*, in the same cluster in *RP* but not in *GP*, in the same cluster in *GP* but not *RP*, and in different clusters in both *RP* and *GP* respectively. When *RP* and *GP* are identical, the index value will be one. When *GP* is only good as a random partition, the index will be zero.

In all our experiments, we set $\varepsilon = 0.001$ and $k=10$ for PCGEN. For fair comparison, multiple values of the parameters required for PROCLUS were tried, and the results with the best accuracy are reported. HARP requires the maximum percentage of outliers as a parameter; this was set to 0 because in sample-based clustering of gene expression data, no data point (sample) is considered as an outlier. The outlier detection mechanism of PROCLUS was also disabled. We standardized the expression values for each gene in the three datasets to mean 0 and variance 1. Table 1 illustrates the ARI values for the three algorithms respectively.

As we can see from Table1, PCGEN is able to achieve highly accurate results and maintain the same performance on the three datasets. With *ALL-AML leukemia* and *Lung cancer*, PCGEN misplaced only one sample. In the case of *MLL leukemia*, our algorithm incorrectly classified only 4 of the 72 samples. These interesting results can be explained by the fact that PCGEN avoids the computation of the distance between samples in full-dimensional spaces. The measure of similarity between different samples is restricted to subsets of attributes where sample values are dense. On the other hand, since our clustering process is based on the *K*-means principle, the accuracy of our algorithm is sensitive to the initial choice of the cluster centers. In order to avoid initialization bias, several initializations are needed.

TABLE 1. ARI VALUES.

	PCGEN	HARP	PROCLUS
ALL-AML Leukemia	0.943	0.631	0.473
MLL Leukemia	0.841	0.526	0.306
Lung cancer	0.978	0.041	0.076

The results of HARP illustrated in Table 1 are less competitive than those of PCGEN. This can be attributed to the fact that the datasets considered here represent some extreme conditions, in which the number of relevant attributes is very low and the dimensionality of the data is very high, misleading HARP's dimension selection procedures. In such situations, the basic assumption of HARP – i.e., that if two data points are similar in high-dimensional space, they have a high probability of belonging to the same cluster in lower-dimensional space – becomes non-obvious.

From Table 1, we remark that the results of PROCLUS are also less accurate than those given by PCGEN. This is because the dimension selection mechanism in PROCLUS, which is based on a distance calculation that involves all dimensions by detecting a set of neighboring objects to a medoid, severely hampers its performance. PROCLUS works best on datasets in which the number of relevant dimensions per cluster is not much lower than the dataset dimensionality.

In order to confirm the effectiveness of our algorithm, we now investigate the importance of dimension selection in the formation of clusters. For this purpose, we calculate the distance ratios DR_1, DR_2 and DR_3, as suggested in [10]. The distance ratios are defined as follows:

$$DR_1(C_S) = \frac{\sum_{x_i \in C_S, A_j \in A_S} (x_{ij} - x_{Sj})^2 / d_S}{\sum_{x_i \in C_S, A_j \in A} (x_{ij} - x_{Sj})^2 / d} \qquad (4)$$

$$DR_2(C_S) = \frac{\sum_{x_i \in C_S, A_j \notin A_S} (x_{ij} - x_{Sj})^2 / (d - d_S)}{\sum_{x_i \in C_S, A_j \in A} (x_{ij} - x_{Sj})^2 / d} \qquad (5)$$

$$DR_3(C_S) = \frac{\sum_{x_i \notin C_S, A_j \in A_S} (x_{ij} - x_{Sj})^2 / d_S}{\sum_{x_i \notin C_S, A_j \in A} (x_{ij} - x_{Sj})^2 / d} \qquad (6)$$

In the above equations, x_{Sj} is the mean of the projected value of cluster C_S on dimension A_j.

DR_1 measures the increase in compactness of the cluster due to dimension selection, DR_2 measures how irrelevant are the non-selected dimensions, and DR_3 measures the increase in separation between the cluster members and other objects due to the selection. For a good cluster, DR_1 should be smaller than 1, DR_2 should be greater than 1, and DR_3 should be larger than DR_1. Tables 2, 3 and 4 illustrate the number of selected genes and the distance ratios of the clusters identified by PCGEN from *ALL-AML leukemia*, *MLL leukemia* and *lung cancer*. The sets *AS* of relevant dimension for each cluster are detected according to Definition 3. For this purpose we set δ =0.8.

TABLE 2. THE DISTANCE RATIOS OF THE TWO CLUSTERS FOUND BY PCGEN FOR *ALL-AML LEUKEMIA*.

Clusters	#genes	DR_1	DR_2	DR_3
ALL	40	0.254	1.004	2.055
AML	20	0.140	1.002	1.421

TABLE 3. THE DISTANCE RATIOS OF THE THREE CLUSTERS FOUND BY PCGEN FOR *MLL LEUKEMIA*.

Clusters	#genes	DR_1	DR_2	DR_3
ALL	91	0.226	1.005	1.279
AML	58	0.262	1.003	1.549
MLL	85	0.293	1.004	1.208

TABLE 4. THE DISTANCE RATIOS OF THE TWO CLUSTERS FOUND BY PCGEN FOR *LUNG CANCER*.

Clusters	#genes	DR_1	DR_2	DR_3
MPM	197	0.152	1.013	1.384
ADCA	239	0.860	1.002	1.024

As illustrated in tables 2, 3 and 4, the number of relevant genes detected by PCGEN for each cluster is extremely low. These results are consistent with the analysis given by Golub et al. [9], in which the authors state that in gene expression data the number of genes which manifest meaningful sample phenotype structure is extremely low. In addition, all the values of the distance ratios DR_1, DR_2 and DR_3 of all the clusters discovered by PCGEN, depicted in tables 2, 3 and 4, satisfy all of the requirements for good clustering. In all three datasets, DR_3 is larger than DR_1; this can be explained by the fact that our algorithm searches for compact and disjoint projected clusters.

On the other hand, the values of DR_2 are just slightly greater than 1; this is because the number of selected genes is extremely low (approximately 1% of all genes) compared to the total number of genes. In other words, in DR_2 we compute the ratio of the distance of the data points from the cluster center in the non-selected dimensions to the distance in the whole data space. In the case where the number of relevant dimensions is extremely low, the value of DR_2 is very close to 1.

Similar behaviours of the distance ratios DR_1, DR_2 and DR_3 were also observed with different values of $\delta \in]0.1, 0.8[$. This can be explained by the fact that the values of the relevance index described in section 3.2 for irrelevant dimensions are close to 0 (equal to 0 in most cases). This is an interesting property of PCGEN, because the parameter δ gives a powerful tool to fine-tune the degree of relevance of an attribute to a cluster.

V. CONCLUSION

We have proposed an efficient distance-based projected clustering algorithm for the challenging problem of clustering samples in gene expression data. Experiments show that PCGEN provides meaningful results and significantly improves the quality of clustering when the dimensionalities of the clusters are much lower than that of the dataset. The accuracy achieved by PCGEN results from the restriction to subsets of attributes imposed on the distance computation, and the initial selection of these subsets. Using this approach, we believe that many distance-based clustering algorithms could be adapted to cluster high-dimensional data sets.

The main focus of our work is to detect projected clusters of extremely low dimensionality embedded in high-dimensional space, so we have not provided a detailed analysis of the biological meaning of each gene selected, which itself is a very important issue. Our algorithm discovers statistically relevant genes, which helps to discriminate between clusters. We believe that each gene selected by our algorithm may characterize some biological phenomenon and can be evaluated using existing biological knowledge or suggest new hypotheses.

Although the results of our algorithm are encouraging, there are some limitations that need be overcome in order to enhance its performance. We plan to propose a more systematic way to set the parameter ε. We also plan to extend the scope of Phase 1 of the proposed algorithm from attribute relevance analysis to attribute relevance and redundancy analysis. This seems to have been ignored by all of the existing projected clustering algorithms.

ACKNOWLEDGMENT

We gratefully thank Kevin Yuk-Lap Yip from Yale University for providing us with the implementations of HARP and PROCLUS used in our experiments. This work is supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, “Automatic Subspace Clustering of High Dimensional Data”, Data Mining and Knowledge Discovery, vol. 11, no. 1, pp. 5-33, 2005.

[2] A. K. Jain, M. N. Mutry, and P.J. Flynn, “Data Clustering: A Review”, ACM Comp. Surveys, vol. 31, no. 3, pp. 264-323, 1999.

[3] K. Beyer, J. Goldstein, R. Ramakrishan, and U. Shaft, “When Is Nearest Neighbor Meaningful?”, Proc. of the 7th International Conference on Database Theory, pp. 217-235, 1999.

[4] H. Liu and L. Yu, “Toward Integrating Feature Selection Algorithms for Classification and Clustering”, IEEE Trans. Knowledge and Data Eng., vol. 17, no. 3, pp. 1-12, 2005.

[5] C.C. Aggarwal, C. Procopiuc, J. L. Wolf, P.S. Yu, and J.S. Park, “Fast Algorithm for Projected Clustering”, Proc. SIGMOD Conf., 1999.

[6] K. G. Woo, J. H. Lee, M. H. Kim, and Y. J. Lee, “FINDIT: A fast and intelligent subspace clustering algorithm using dimension voting”, Information and Software Technology, vol. 46, no. 4, pp. 255-271, 2004.

[7] G. Getz, E. Levine, and E. Domany, “Coupled Two-Way Clustering Analysis of Gene Microarray Data”, Proc. Nat’l Academy of Science, vol. 97, no 22, pp. 12079-12084, Oct. 2000.

[8] D. Jiang, C. Tang and A. Zhang, “Cluster Analysis for Gene Expression Data: A Survey”, IEEE Trans. Knowledge and Data Eng., vol. 16, no. 11, pp 1370-1386, 2004.

[9] T. R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Meisirov, H. Coller, M.L. Loh, J.R. Downing, M. A. Caligiuri, C.D. Bloomfield, E.S. Lander, “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring”, Science, vol. 286, no. 15, pp. 531-537, 1999.

[10] K.Y. Yip, D.W. Cheng, M.K. Ng and K. Cheng, “Identifying Projected Clusters From Gene Expression Profiles”, Journal of Biomedical Informatics, vol. 37, no. 5, pp. 345-357, 2004.

[11] C.C. Aggarwal and P.S. Yu, “Redefining Clustering for High Dimensional Applications”, IEEE Trans. Knowledge and Data Eng., vol. 14, no. 2, pp 210-225, 2002.

[12] K.Y. Yip, D.W. Cheng and M.K. Ng, “HARP: A Practical Projected Clustering Algorithm”, IEEE Trans. Knowledge and Data Eng., vol. 16, no. 11, pp 1387-1397, 2004.

[13] .Y. Yip, D.W. Cheng and M.K. Ng, “On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering”, Proc. ICDE, pp. 329-340, April 2005.

[14] E. K. K. Ng, A. W. F, and R C. W. Wong, “Projective Clustering by Histograms”, IEEE Trans. Knowledge and Data Eng., vol. 17, no. 3, pp. 369-383, 2005.

[15] L. Parsons, E. Haque, and H. Liu, “Subspace Clustering for High Dimensional Data: A Review”, ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90-105, 2004.

[16] L. Hubert and P. Araie, “Comparing Partitions”, Journal of Classificastion, vol. 2, pp. 193-218, 1985.

[17] G. W. Milligan and G.W. Cooper, “A Study of the comparability of External Criteria for Hiearchical Cluster Analysis”, Multivariate Behavior Research, vol. 21, pp. 441-458, 1986.

[18] D. Steinley, “Properties of the Hubert-Arabie Adjusted Rand Index”, Psychological Methods, vol. 9, no. 3, pp. 386-396, 2004.