Adaptive dimension reduction for clustering high dimensional data

Chris Ding[a], Xiaofeng He[a], Hongyuan Zha[b] and Horst D. Simon[a]

[a] NERSC Division, Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720
[b] Department of Computer Science and Engineering
Pennsylvania State University, University Park, PA 16802
{chqding,xhe,hdsimon}@lbl.gov, zha@cse.psu.edu

Abstract

It is well-known that for high dimensional data clustering, standard algorithms such as EM and the K-means are often trapped in local minimum. Many initialization methods were proposed to tackle this problem , but with only limited success. In this paper we propose a new approach to resolve this problem by repeated dimension reductions such that K-means or EM are performed only in very low dimensions. Cluster membership is utilized as a bridge between the reduced dimensional subspace and the original space, providing flexibility and ease of implementation. Clustering analysis performed on highly overlapped Gaussians, DNA gene expression profiles and internet newsgroups demonstrate the effectiveness of the proposed algorithm.

1 Introduction

In many application areas, such as information retrieval, image processing, computational biology and global climate research, analysis of high dimensional datasets is frequently encountered. For example, in text processing, typical dimension of a word vector is of the size of the vocabulary of a document collection and tens of thousands of words/phrases are used routinely; in molecular biology, human gene expression profile analysis typically involves thousands of genes; and in image processing, a typical 2-dim image has $128^2 = 16{,}384$ pixels or dimensions.

Developing effective clustering methods to handle high dimensional dataset is a challenging problem. Popular clustering methods such as the K-means and EM methods suffer from the well-known local minima problem: as iterations proceed, they are often trapped in the local minima in the configuration space, due to the greedy nature of these algorithms. In high dimensional space, the equi-potential (cost function) surface is very rugged. The iterations almost always get trapped somewhere close to the initial starting configuration. In other words, it is difficult to sample through a large configuration (parameter) space. The conventional approach is to do a large number of runs with random initial starts and pick up the best one as the result [24, 26]. Besides random starts, there are a number of initialization methods, most of which concentrate on how to intelligently choose the starting configurations (the K centers) in order to be as close to the global minima as possible [5, 25, 22, 17]. However, these approaches are limited by the inherent difficulty of finding global minima in high dimensional space in the first place. Monte Carlo methods are also used [28].

In this paper, we propose a new approach to solve this problem. Our approach utilizes the idea of dimension reduction. Dimension reduction is often used in clustering, classification, and many other machine learning and data mining applications. It usually retains the most important dimensions (attributes), removes the noisy dimensions (irrelevant attributes) and reduces computational cost.

In most applications, dimension reduction is carried out as a *preprocessing* step. The selection of the dimensions using principal component analysis (PCA) [20, 14] through singular value decomposition (SVD) [15] is a popular approach for numerical attributes. In information retrieval, latent semantic indexing uses SVD to project textual documents represented as document vectors [7]; SVD is shown to be the optimal solution for a probablistic model for document/word occurrence [12]. Random projections to subspaces have also been used [13, 6].

In all those applications, however, once the dimensions are selected, they stay fixed during the entire clustering process. The dimension reduction process is decoupled from the clustering process. If the data distribution is far from Gaussian, for example, the dimensions selected using PCA may deviate substantially from the optimal.

Here (i) we approach dimension reduction as a *dynamic* process that should be adaptively adjusted and integrated with the clustering process; (ii) we make effective use of cluster membership as the bridge connecting the clusters defined in the reduced dimensional space (subspace) and those defined in the full dimensional space; (iii) using this connection, clusters are discovered in the low dimensional subspace to avoid the curse of dimensionality [27] and are adaptively re-adjusted in the full dimension space for global optimality. This process is repeated until convergence.

In this paper we focus on the K-means and EM algorithms using the mixture model of spherical Gaussian components. Using marginal distributions, the gaussian mixtures retain identical model parameters in reduced low-dimensional subspace as in the original high dimensional space, providing a theoretical justification for dimension reduction. The objective function for the K-means has the same property.

K-centroid classification on text are studied via dimension reduction in [18], where K centroids are used to define the subspace projection. Dimension reduction in text processing has been extensively studied [4, 12, 9, 21]. All of above studies use dimension reduction as preprocessing; while in our approach, dimension reduction is performed adaptively.

In this paper, we consider projection methods in which the new projected dimensions are linear combination of old dimensions. Optimal selection of a subset of existing dimensions (attributes) is a substantially different approach. Selection of a subset of attributes in the context of clustering are studied in [2, 1]. In the context of classification, subset selection has been extensively studied [19].

2 Effective Dimension for Clustering

Our approach is to perform clustering in low dimensional subspaces. EM, in essence, is fitting a density functional form and is sensitive to local density variations. In the much reduced-dimension subspaces, we have a much smoother density [27], therefore reducing the chances of being trapped in the local minima.

We may interpret the low dimensional subspace as containing the relevant attributes (linear combinations of coordinates). What is the dimensionality r of the reduced-dimension subspace for clustering? We argue that $r \leq K-1$ based on linear discriminant analysis: Given two gaussian distributions with means μ_1, μ_2, and pooled covariance matrix Σ in d-dim space, a point x is classified to belong to class 1 or 2 depending upon

$$x^T \Sigma^{-1}(\mu_1 - \mu_2) \geq \text{threshold}$$

For spherical Gaussian, $\Sigma = \sigma^2 I$ (I is the identity matrix), thus the $d-1$ subspace perpendicular to the direction $\mu_1 - \mu_2$ does not enter the consideration. The effective dimension for 2 clusters is 1. For $K > 2$, we may consider all pairs of two-class classifications, and the dimensions perpendicular to all $\mu_i - \mu_j$ directions ($i \neq j$) are irrelevant. Thus the effective clustering dimensions for the K spherical gaussians are spanned by the K centers $\mu_1, \cdots, \mu_K$, for which $r = K-1$. We call the relevant dimensions passing through all K centers the r-dim subspace. This conclusion is essentially a geometric point of view. If Euclidean distances are the main factors in clustering, the dimensions perpendicular to the relevant subspace are clearly irrelevant.

The effective dimensionality of the relevant subspace could be less than $K-1$. This happens when the K cluster centers lie in a subspace with dimension $r < K-1$. For example, there could be 5 clusters with all their centroids lie on a 2-dim plane. In summary, the effective dimension for clustering is $r \leq K-1$.

3 EM in relevant subspace

Our algorithm can be easily and naturally incorporated into Expectation-Maximization (EM) algorithm [8, 23] applied to spherical Gaussian mixtures. The idea is that the irrelevant dimensions can be integrated out, and the resulting marginal distribution follows the same Gaussian mixture functional form. Then we can freely move between the reduced-dimension subspace and the original space. In this approach, cluster membership information (posterior probabilities of the indicator variables) plays a critical role. Knowing them in the reduced-dimension subspace we can directly infer the centers in the original space. We assume the following mixture model

$$p(\mathbf{x}) = \pi_1 g_1^d(\mathbf{x} - \mu_1) + \cdots + \pi_K g_K^d(\mathbf{x} - \mu_K) \qquad (1)$$

where each component is a spherical Gaussian distribution,

$$g_k^d(\mathbf{x}) = \frac{1}{(\sqrt{2\pi}\sigma_k)^d} \exp(-\frac{\|\mathbf{x} - \mu_k\|^2}{2\sigma_k^2})$$

and $\mathbf{x}, \mu_k$ are vectors in d-dim space. We denote it as $N^{(d)}(\mu_k, \sigma_k)$. Note that the spherical gaussian function has two invariant properties that will be important later on: (i) they are invariant under any orthogonal coordinate rotation operation $R : \mathbf{x} \to R\mathbf{x}$:

$$g_k^d(R\mathbf{x}|R\theta) = g_k^d(\mathbf{x}|\theta)$$

where R is a $d \times d$ orthonormal matrix and $\theta = \{\mu_1, \cdots, \mu_K, \sigma_1, \cdots, \sigma_K\}$ are the model parameters. (ii) they are invariant under coordinate translation (shift) operation $L : \mathbf{x} \to \mathbf{x} + \ell$: we have

$$g_k^d(L\mathbf{x}|L\theta) = g_k^d(\mathbf{x}|\theta)$$

Given the gaussian mixture model, dimension reduction can be properly studied in a probabilistic framework using marginal distributions. For this reason, we need to split the space into an r-dim space which contains all the relevant dimensions (attributes), and an s-dim space ($s = d - r$) which contains all the irrelevant dimensions (noises). We split the coordinates into $\mathbf{y} = R^T\mathbf{x} = (R_r, R_s)^T\mathbf{x}$, or more explicitly,

$$\begin{pmatrix} \mathbf{y}^{\parallel} \\ \mathbf{y}^{\perp} \end{pmatrix} = \begin{pmatrix} R_r^T\mathbf{x} \\ R_s^T\mathbf{x} \end{pmatrix}, \quad \begin{pmatrix} \nu^{\parallel} \\ \nu^{\perp} \end{pmatrix} = \begin{pmatrix} R_r^T\mu \\ R_s^T\mu \end{pmatrix} \tag{2}$$

where $\mathbf{y}^{\parallel}, \nu^{\parallel}$ are in r-dim relevant space, and $\mathbf{y}^{\perp}, \nu^{\perp}$ are in s-dim subspace of noise, orthogonal to the r-dim relevant space. R is the coordinate rotation, such as the coordinate transformation used in PCA, to clearly separate those relevant and noisy dimensions.

The marginal distribution is defined as

$$p(\mathbf{y}^{\parallel}) \equiv \int p(\mathbf{x}) J d\mathbf{y}^{\perp} = \int p(\mathbf{y}^{\parallel}, \mathbf{y}^{\perp}) J d\mathbf{y}^{\perp}.$$

where $J = \det(\partial\mathbf{x}/\partial\mathbf{y}) = \det(R)$ is the Jacobian related to coordinate transformation. For orthonormal rotations such as U and V in PCA, $R^T R = R^T R = I_k$ and $\det(R) = 1$. Splitting coordinates, we have

$$\begin{aligned} \|\mathbf{x} - \mu\|_d^2 &= \left\| \begin{bmatrix} R_r^T(\mathbf{x} - \mu) \\ R_s^T(\mathbf{x} - \mu) \end{bmatrix} \right\|_d^2 \\ &= \|\mathbf{y}^{\parallel} - \nu^{\parallel}\|_r^2 + \|\mathbf{y}^{\perp} - \nu^{\perp}\|_s^2 \end{aligned} \tag{3}$$

Thus we have $g_k^d(\mathbf{x} - \mu) = g_k^r(\mathbf{y}^{\parallel} - \nu_k^{\parallel}) \cdot g_k^s(\mathbf{y}^{\perp} - \nu_k^{\perp})$. The marginal distribution of $g_k^d(\mathbf{x})$ becomes

$$\int g_k^d(\mathbf{x} - \mu) d\mathbf{y}^{\perp} = g_k^r(\mathbf{y}^{\parallel} - \nu^{\parallel}), \tag{4}$$

which is exactly the standard spherical Gaussian in the r-dim subspace. For this reason, we simply use $\mathbf{y}$ for $\mathbf{y}^{\parallel}$ and ν for $\nu^{\parallel}$ in the r-dim subspace. Therefore we conclude that

Theorem 1. In EM clustering using spherical Gaussian mixture models in d-dimensions, after integrating out irrelevant dimensions, the marginal probability becomes

$$p(\mathbf{y}) = \pi_1 g_1^r(\mathbf{y} - \nu_1) + \cdots + \pi_K g_K^r(\mathbf{y} - \nu_K), \tag{5}$$

exactly the same type of Gaussian distribution as in r-dim space. All relevant attributes for clustering are retained in the r-dim subspace.

4 Adaptive Dimension Reduction for EM

For real-world clustering problems where clusters are not well-separated, the r-dim subspace initially obtained using PCA does not necessarily coincide with the subspace spanned by the K cluster centers. Therefore, the centers, and cluster memberships, in the usual dimension reduction clustering are not necessarily the correct (or accurate) results. One can correct this by adaptively modifying the r-dim subspace using the most current clustering results, and do another round of clustering in the modified subspace. One can repeat this process several times to improved the results.

Given a point or a cluster centroid in the r-dim subspace, mapping back to the original d-dim space is not *unique*. In fact, there are infinite number of points in the d-dim space, all of which project into one point in the r-dim subspace (all points on a vertical line project into a single point in x-y plane). However, the centers (or centroids in the K-means) obtained in clustering in the r-dim subspace can be *uniquely* traced back to the original d-dim space by using the cluster membership of each data point. This observation is the basis of our ADR-EM clustering.

The cluster membership information is contained in the posterior probability h_i^k,

$$h_i^k = \Pr(c_i = k | y_i, \theta)$$

the probability of point i belongs to cluster c_k given current model (parameters) and the evidence (value of y_i). EM algorithm is the following: (i) initialize model parameters $\{\pi_k, \nu_k, \sigma_k\}$; (ii) compute $\{h_i^k\}$, $h_i^k = \pi_k g_k^r(y) / \sum_k \pi_k g_k^r(y)$; (iii) update: (1) compute the number of points belonging to cluster c_k: $n_k = \sum_i h_i^k$; (2) update priors: $\pi_k = n_k/N$; (3) update centers: $\nu_k = \sum_i h_i^k y_i / n_k$; (4) update covariances: $\sigma_k = \sum_i h_i^k ||y_i - \mu_k||^2 / r n_k$. Steps (ii) and (iii) are repeated until convergence. Once EM converges, the final cluster information is contained in $\{h_i^k\}$. Using this information, the centers in the original d-dim space can be computed as

$$\mu_k = \sum_i h_i^k x_i / n_k.$$

Once the locations of the K cluster centers in the original d-dim space are known, expressed as the $d \times K$ matrix

$$C_{\scriptscriptstyle K} = (\mu_1, \mu_2, \cdots, \mu_{\scriptscriptstyle K}), \tag{6}$$

and we can easily find the new $r = K - 1$ dimensional subspace spanned by these K centers. The new subspace is defined by a set of $K-1$ orthonormal vectors $U_r = (u_1, u_2, \cdots, u_r)$. Note that this orthonormal basis is not unique: any rotation of it is an equally good basis. Here we present two methods to compute the basis. Both have the same $O(r^2 \cdot N)$ computational complexity where N is the number of data vectors in the dataset.

4.1 SVD Basis

We compute the singular value decomposition (SVD) [15] of C as: $C = \sum_\ell \mathbf{u}_\ell \lambda_\ell \mathbf{v}_\ell^T$. Since the data is centered, $\sum_k \pi_k \mu_k = \sum_i \mathbf{x}_i / n = 0$, $\{\mu_1 \cdots \mu_{\scriptscriptstyle K}\}$ are linearly dependent. Therefore C has rank $r = K - 1$. The $d \times r$ matrix $U_r = (u_1, u_2, \cdots, u_r)$ is the orthogonal basis of the new subspace. The SVD basis has a useful property that it automatically orders the dimensions according to their importance. For example, the last dimension u_r is not as important as the first sub-dimension u_1, as in principal component analysis.

Now we project the original data into the new subspace using $y_i = U_r^T x_i$ and do another round of EM clustering, with the starting cluster centers from the projections $\nu_i = U_r^T \mu_i$ and information on priors $\{\pi_k\}$.

4.2 QR basis

Another way to build the orthogonal basis is to use QR or Gram-Schmidt on the K centroids [18]. Without loss of generality, we let $\mu_{\scriptscriptstyle K}$ be the one with the smallest magnitude, and form a $d \times (K-1)$ matrix

$$C_{\scriptscriptstyle K-1} = (\mu_1 - \mu_{\scriptscriptstyle K}, \mu_2 - \mu_{\scriptscriptstyle K}, \cdots, \mu_{\scriptscriptstyle K-1} - \mu_{\scriptscriptstyle K}). \tag{7}$$

Note that vectors in C_r are not necessarily orthogonal to each other. We use the Gram-Schmidt procedure, or equivalently, the QR factorization in linear algebra,

$$C_{k-1} = Q \begin{bmatrix} P \\ 0 \end{bmatrix} = [Q_r, Q^\perp] \begin{bmatrix} P \\ 0 \end{bmatrix} = Q_r P,$$

where the $d \times r$ matrix $Q_r = (\mathbf{q}_1, \cdots, \mathbf{q}_r)$ is the orthonormal basis for the subspace. P is an $r \times r$ upper triangle matrix, containing the projections of components in the C_r basis. This QR basis has the property that $\mathbf{q}_k$ will be close to the μ_k centers if they are reasonably orthogonal to each other.

Now we use Q_r to project the original data into the new subspace by $y_i = Q_r^T x_i$, etc. Note that by construction, no centers can coincide with each other in either SVD or QR basis.

4.3 The complete ADR-EM algorithm

The complete Adaptive Dimension Reduction Expectation Maximization (ADR-EM) algorithm is described as follows.

1. Preprocessing data to fit better the spherical Gaussian model. Center the data such that $\sum_i \mathbf{x}_i = 0$. Rescale the data such that the variance in every dimension is 1. Choose appropriate K as input parameter. Choose dimensionality r for the reduced dimension subspace. In general, we recommend $r = K - 1$. But $r = K$ or $r < K - 1$ are also appropriate.

2. Do the first dimension reduction using PCA or any other methods, including random starts.

3. Run EM in the r-dim subspace to obtain clusters. Use cluster membership to construct cluster centroids in the original space. Check convergence. If yes, go to step 5.

4. Compute the new r-dim subspace spanned by the K centroids using either SVD or QR basis. Project data into this new subspace. Go to step 3.

5. Output results and converting posterior probabilities to discrete indicators. The relevant attributes (coordinates) are also identified.

If accurate results are necessary, one may run one final round of EM in the original data space starting with existing parameters (see section 7).

A key feature of ADR is that no matter how the data are projected and transformed (shifted, rotated, etc) in subspaces, once the cluster memberships in the subspace are computed, we can always use them to construct clusters in the original space, no need for bookkeeping of the details of data transformations and/or reductions. One can easily design hybrid schemes of different data projections and use the obtained cluster membership as the bridge between them to form an integrated clustering method.

4.4 Relevant dimensions

In general, $r = K-1$ is the optimal choice. However, $r = K$ is also a good choice in many cases: (i) when using QR basis, the QR can be applied to C_K (cf. Eq.6) instead of C_{K-1} (cf. Eq.7) and obtain K basis vectors; (ii) in either SVD basis or QR basis, we can add one or even more additional basis vectors which are orthogonal to existing basis. These additional basis vectors can be either chosen for a particular emphasis or chosen randomly. Randomly choosing additional basis vector could help to search for broader configuration space, making sure we are not stuck in a local minimum.

Sometimes we can also choose $r < K-1$. Although K centers define a $(K-1)$-dim subspace, they can sometimes locate on or near an r-dim subspace where $r < K-1$. For example, 4 points in a 3-dim space could lie on a 2-dim plane or even on a 1-dim line. In these cases, C is rank deficient, i.e., the rank of C will be less than $K-1$ and the singular values in SVD basis will drop to near zero; we should choose the appropriate $r < K-1$.

Even if C is not rank deficient, we may still set r to be less than $K-1$ for computational efficiency and effectiveness. This is especially important if we are dealing with a large and complex dataset and somehow we believe there should be, for example, $K = 10$ clusters. Due to the curse of dimensionality, 9-dim space may still be too high, so we may set $r = 3$ and find 10 clusters in 3-dim space where EM or K-means are typically more effective. Also in 3-dim space, computation is more efficient (than in 9-dim) and the results can be inspected using 3-dim graphics or other visualization tools. In this case, after the best 10 clusters are discovered using $r = 3$, we may further refine the results by setting $r = 9$ and re-run the algorithm, using cluster membership as the bridge.

In all the test examples below, we have tested this $r < K-1$ (over-reduced) method and the results are generally the same as the $r = K-1$ method. However, we do notice the slower convergence of the EM method.

5 Adaptive Dimension Reduction for K-means

The ADR method can also be applied to the K-means clustering as well. Given a set of data vectors $X = [x_1, \ldots, x_n]$, the K-means for K clusters seeks to find a set of centers $C_\mu = [\mu_1, \ldots, \mu_K]$ so as to minimize

$$J_d(X, C_\mu) = \sum_{k=1}^{K} \sum_{i \in c_k} \|x_i - \mu_k\|_d^2. \qquad (8)$$

Each cluster c_i is represented by a center μ_i and consists of the data vectors that are closest to it in Euclidean distance, and the center of a cluster is the centroid of its data vectors. The K-means clustering can be viewed as a special case of EM with three simplifications (i) $\sigma_1 = \cdots = \sigma_K = \sigma$; (ii) $\pi_1 = \cdots = \pi_K$; (iii) with $\sigma \to 0$ so that $h_i^k = 1$ or 0.

As before, the key is to find the relevant r-dim reduced space, specified by the projection matrix R_r. We have the following.

Theorem 2. Suppose we somehow know the correct r-dim relevant subspace defined by R_r. Let $Y = R_r^T X = R_r^T(x_1, \ldots, x_n)$ and $C_\nu = [\nu_1, \ldots, \nu_K]$ be K centroids in r-dim subspace. Solve the K-means problem in r-dim subspace,

$$\min_{C_\nu} J_r(Y, C_\nu).$$

Use the cluster membership $H = (h_i^k)$ obtained to reconstruct the K centers $C_\mu^* = [\mu_1^*, \ldots, \mu_K^*]$ in the full dimensional space. Then C_μ^* are the exact optimal solution to the the full-dimension K-means problem.

Proof. Assume the centroid matrix C_μ^* are the minimum for K cluster K-means in (8). Construct projection matrix R_r that spans the subspace for C_μ^*. Use Gram-Schmit procedure to construct R_s such that $R = [R_r, R_s]$ is orthonormal matrix. Since R_r spans the subspace for C_μ^*, $R_s^T \mu_k^* = 0$. Using this fact and Eq.(2), we have

$$\|x_i - \mu_k^*\|_d^2 = \|y_i - \nu_k^*\|_r^2 + \|R_s^T x_i\|_s^2$$

This indicates that among all K centers, if c_k^* is closest to x_i in the d-sim space, then $R_r^T c_k^*$ is closest to $R_r^T x_i$ in the r-dim space, independent of R_s. We can write $J_d(X, C) = J_r(Y, C_\nu) + \text{const}$. □

If we know the final solution C_μ^*, we can easily construct R_r. For any $r \times r$ orthonormal matrix S, $R_r S$ still spans the correct subspace. In practice we do not know C_μ^* until after the problem is solved. By Theorem 2, we only need to find the relevant subspace. Because of the large flexibility in defining R_r, finding the relevant subspace is much easier than finding C_μ^* directly. This is the usefulness of Theorem 2. Our adaptive dimension reduction K-means is based on the theorem. The complete ADR-Kmeans algorithm is identical to ADR-EM algorithm in §4.3.

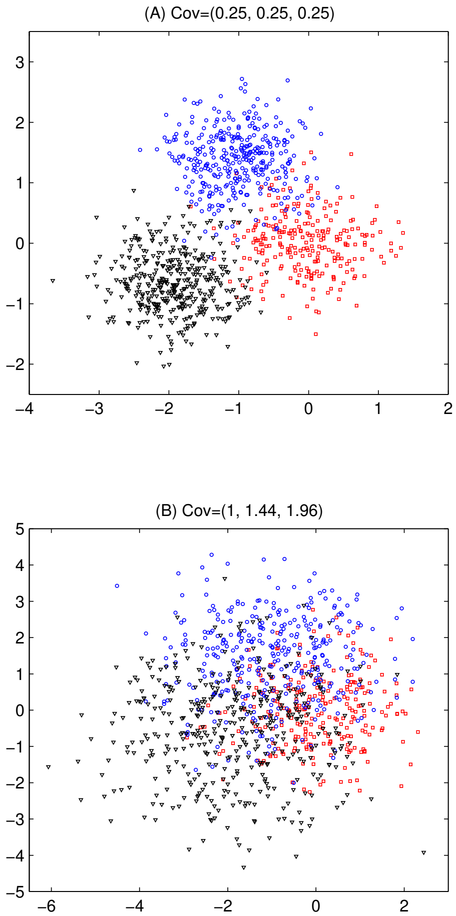

Figure 1: Three Gaussian clusters in 4-dim space. (A) covariances (0.25, 0.25, 0.25). (B) covariances (1, 1.44, 1.96). Data points are shown in the first 2 PCA components. From (A) to (B), variance increases $(1/0.5)^2$ for the 1st cluster (red squares), $(1.2/0.5)^2$ for the 2nd cluster (blue circles), and $(1.4/0.5)^2$ for the 3rd cluster (black triangles).

6 Applications

6.1 Highly overlapping Gaussian mixtures

The first example is a 1000-point synthetic dataset of 3 gaussians in 4-dim with centers $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, listed below

$$(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \quad = \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{9}$$

covariances $(\sigma_1^2, \sigma_2^2, \sigma_3^2) = (1, 1.44, 1.96)$, and prior distributions $(\pi_1, \pi_2, \pi_3) = (0.25, 0.35, 0.4)$. The 3 gaussians are highly overlapped (see Fig.1) The results of ADR-EM are shown below,

$$(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*) = \begin{bmatrix} 0.03 & 0.06 & 1.04 \\ 0.23 & 1.18 & 0.99 \\ -0.03 & 1.02 & -1.32 \\ 0.12 & 1.16 & 1.00 \end{bmatrix}$$

quite close to the correct results. Repeated runs show that the method is quite robust. If we run EM directly in 4-dim space, the EM will have difficulty finding the correct clusters. Results change for each different run. After 10 runs, the best results are shown below

$$(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*) = \begin{bmatrix} 0.38 & 0.12 & 1.44 \\ 0.20 & 0.97 & 1.76 \\ -0.69 & 0.72 & -1.96 \\ 0.30 & 0.91 & 1.38 \end{bmatrix}$$

Dimension reduction is essential in this highly-overlapped situation.

6.2 DNA Microarray gene expression profiling

This example is from molecular biology. High density DNA microarray technology can simultaneously monitor the expression level of thousands of genes which determines different pathological states of the same tissue drawn from different patients [16, 3]. Here we study gene expression profiles of non-Hodgkin lymphoma cancer data from [3]. Among the 96 samples of 9 phenotypes (classes), we pick the 4 largest classes with a total of 76 samples(see Fig.2): (1) 46 samples of diffuse large B-cell lymphoma ($\circ$), (2) 10 samples of Activated Blood cell B-cell ($\bigtriangledown$), (3) 9 samples of chronic lymphocytic leukemia ($\bigtriangleup$), (4) 11 samples of follicular lymphoma (+).
Each sample contains expression levels of 4026 genes (variables). The question we ask: could we discover these phenotypes from data directly, without human expertise?

We use t-test statistic criteria to select top 100 genes. The clustering problem is focused on the 76 samples in 100-dim space with K=4. This is still a high dimensional problem. We use ADM-EM algorithm on this dataset, setting r=3. The clustering result is shown in the following contingency table

$$T = \begin{bmatrix} 39 & 3 & 4 & \cdot \\ \cdot & 10 & \cdot & \cdot \\ \cdot & \cdot & 9 & \cdot \\ \cdot & \cdot & \cdot & 11 \end{bmatrix}$$

where $T = (t_{ij})$, t_{ij} is the number of data points which are observed to be in cluster i, but was computed via the clustering method to belong to cluster j. The accuracy is 69/76=0.91% (accuracy is defined as $\sum_k t_{kk}/N$ [11]). If we perform the clustering directly in the 100-dim space, the runs are often trapped in local minimum. The usefulness of PCA on gene expression analysis were noted in [10].

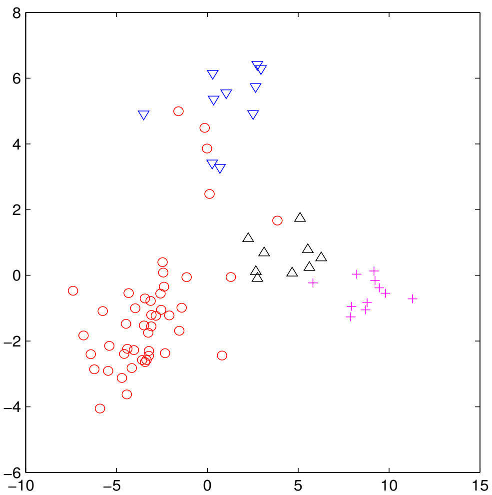

Figure 2: Gene expression profiles for lymphoma cancer dataset. Shown are the data in the first 2 PCA components.

6.3 Internet newsgroups clustering

We use the Internet newsgroups dataset [1] to illustrate the process of adaptive dimension reduction. We use five news groups NG2/NG9/NG10/NG15/NG18 with 50 news articles from each group (see [29] for details). `NG2: comp.graphics; NG9: rec.motorcycles; NG10: rec.sport.baseball; NG15: sci.space; NG18: talk.politics.mideast.`

Words with document frequency less than 3 are removed, and a total of 2731 distinct words are retained. Each document is represented by a vector in this d=2731 dimensional space. We set $r = 5$ (relevant dimension is a 5-dim subspace). We start with a *random* initial 5-dim subspace. In Table 1, we list the accuracy and J at the end of each adaptive iteration. Repeated adaptive dimension reduction gradually converges to the correct subspace. As a comparison, we run K-means algorithm in the original d=2731 space with the same initial clusters and obtain accuracy 50.40% and J=227.83. This indicates the effectiveness of our adaptive method.

[1]The newsgroups dataset together with the `bow` toolkit for processing it can be downloaded from http://www.cs.cmu.edu /afs/cs/project/theo-11/www/naive-bayes.html.

iteration	accuracy	min(J)
1	25.60	235.58
2	34.40	233.11
3	44.80	231.01
4	52.80	229.68
5	60.40	228.19
6	63.60	227.47
7	66.00	227.06
8	67.20	226.86

Table 1: Clustering results of ADM-Kmeans algorithm.

7 Discussions

We introduced a new method for clustering high dimensional data using adaptive dimension reductions. The key to the effectiveness of this method lies in (Theorems 1 and 2) that working in the subspace containing true cluster centers is sufficient to find the cluster centers. The subspace containing cluster centers is of dimension K, far smaller than the original dimension in many applications. Adaptive dimension reduction is an effective way to converge to this subspace. Note that finding the subspace is much easier than finding cluster centers directly, due to the flexibility in defining subspace.

Although we concentrate on EM and K-means algorithm here, the adaptive dimension reduction approach could be extended to other clustering methods. Using cluster membership as the bridge to connect subspaces of different dimensions makes these extensions easy to implement. For example, one may construct a number of subspaces based on different feature selection methods and apply different clustering methods on these subset of features and move or combine them to satisfy some optimal conditions.

Another interesting subtle point is that although the functional form in d-dim space [cf. Eq.(1)] is very much the same as that in r-dim subspace [cf. Eq.(5)], the final parameters are *not* the same: the priors π_k differ in the two spaces in the case of highly overlapped clusters with different covariances. The reason is that the probability can not be separated into a product of P(relevant coordinates)*P(irrelevant coordinates): $p(\mathbf{x}) = p(\mathbf{y}, \mathbf{y}^{\perp}) \neq p(\mathbf{y}) \cdot p(\mathbf{y}^{\perp})$, even if each mixture component is separable. Therefore, the standard practice of reporting the results directly obtained in the reduced-dimension subspace is not accurate enough. For this reason, we suggest the EM in the d-dim space be run once using those parameters obtained in the r-dim subspace to get more accurate final parameters.

Acknowledgments. This work is supported by U.S. Department of Energy (Office of Science, Office of Advanced Scientific Research, MICS Division, and Office of Laboratory Policy and Infrastructure) under contract DE-AC03-76SF00098.

References

[1] C.C. Aggarwal, C. Procopiuc, J.L. Wolf, P.S. Yu, and J.S. Park. Fast algorithms for projected clustering. pages 61–72, 1999.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. pages 94–105, 1998.

[3] A.A. Alizadeh, M.B. Eisen, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[4] M.W. Berry, S.T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[5] P. Bradley and U. Fayyad. Refining initial points for k-means clustering. In *Proc. 15th International Conf. on Machine Learning*, 1998.

[6] S. Dasgupta. Experiments with random projection. In *Proc. 16th Conf. Uncertainty in Artificial Intelligence (UAI 2000)*, 2000.

[7] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Info. Sci*, 41:391–407, 1990.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via em algorithm. *J. Royal Stat. Soc. B.*, pages 1–38, 1977.

[9] I. Dhillon and D. Modha. Concept decomposition for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[10] C. Ding. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proc. 6th Int'l Conf. Research in Comp. Mol. Bio.(RECOMB 2002)*, pages 127–136, April 2002.

[11] C. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349–358, 2001.

[12] C.H.Q. Ding. A similarity-based probability model for latent semantic indexing. *Proc. 22nd ACM SIGIR Conference*, pages 59–65, Aug. 1999.

[13] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proc. 19th ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd ed.* Wiley, 2000.

[15] G. Golub and C. Van Loan. *Matrix Computations, 3rd edition.* Johns Hopkins, Baltimore, 1996.

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

[17] J. Grim, J Novovicova, P. Pudil, P. Somol, and F. Ferri. Initialization normal mixtures of densities. *Proc. Int'l Conf. Pattern Recognition (ICPR 1998)*, Dec. 1998.

[18] M. Jeon, H. Park, and J.B. Rosen. Dimension reduction based on centroids and least squares for efficient processing of text data. *Proc. SIAM Conf. on Data Mining*, 2001.

[19] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Int'l Conf. Machine Learning*, pages 121–129, 1994.

[20] I.T. Jolliffe. *Principal Component Analysis.* Springer Verlag, 1986.

[21] G. Karypis and E.-H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. *Proc. 9th Int'l Conf. Information and Knowledge Management (CIKM 2000)*, 2000.

[22] R. Kothari and D. Pitts. On finding the number of clusters. *Pattern Recognition Letters*, 50:405 –416, 1999.

[23] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley, 1997.

[24] M. Meila and D. Heckerman. An experimental comparison of several clustering and initialization methods. *Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI 98)*, pages 386–395, 1998.

[25] A.W. Moore. Very fast em-based mixture model clustering using multiresolution kd-trees. *Proc. Neural Info. Processing Systems (NIPS 1998)*, Dec. 1998.

[26] J. Pena, J. Lozano, and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 50:1027–1040, 1999.

[27] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley, 1992.

[28] C.K.I. Williams. A MCMC approach to hierarchical mixture modeling. *Proc. Neural Info. Processing Systems (NIPS 2000)*, Dec. 2000.

[29] H. Zha, C. Ding, M. Gu, X. He, and H.D. Simon. Spectral relaxation for k-means clustering. *Proc. Neural Info. Processing Systems (NIPS 2001)*, Dec. 2001.