Discriminative Cluster Analysis

Fernando De la Torre FTORRE@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University. 5000 Forbes Av., Pittsburgh, PA 15213 USA

Takeo Kanade TK@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University. 5000 Forbes Av., Pittsburgh, PA 15213 USA

Abstract

Clustering is one of the most widely used statistical tools for data analysis. Among all existing clustering techniques, k-means is a very popular method because of its ease of programming and because it accomplishes a good trade-off between achieved performance and computational complexity. However, k-means is prone to local minima problems, and it does not scale well with high dimensional data sets. A common approach to dealing with high dimensional data is to cluster in the space spanned by the principal components (PC). In this paper, we show the benefits of clustering in a low dimensional discriminative space rather than in the PC space (generative). In particular, we propose a new clustering algorithm called Discriminative Cluster Analysis (DCA). DCA jointly performs dimensionality reduction and clustering. Several toy and real examples show the benefits of DCA versus traditional PCA+k-means clustering. Additionally, a new matrix formulation is suggested and connections with related techniques such as spectral graph methods and linear discriminant analysis are provided.

1. Introduction

Clustering is one of the most widely used statistical methods in data analysis (e.g. multimedia content-based retrieval, molecular biology, text mining, bioinformatics). Recently, with an increasing number of database applications that deal with very large high dimensional datasets, clustering has emerged as a very important research area in many disciplines. Unfortunately, many known algorithms tend to break down in high dimensional spaces because of the sparsity of the points. In such high dimensional spaces not all the dimensions might be relevant for clustering, outliers are more difficult to detect, and it is not necessarily clear which distance measure to choose. On the other hand, when handling large amounts of data, time complexity can becomes a limiting factor.

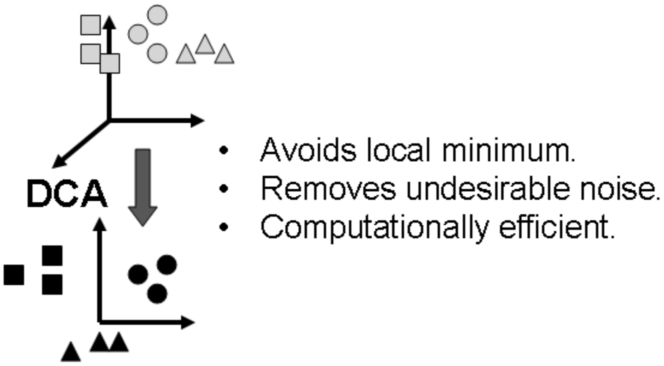

Figure 1. DCA finds a low dimensional projection beneficial for clustering.

Two types of clustering algorithms exist: partitional and hierchical (Jain et al., 1999). Partitional methods (e.g. k-means, mixture of Gaussians, graph theoretic, mode seeking) produce just one partition of the data; whereas, hierarchical ones (e.g single link, complete link) produce several of them. In particular, k-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that has been extensively studied and extended (Jain, 1988). Although being an extremely popular technique because of its ease of programming and performance in large high dimensional data sets, k-means suffers from many drawbacks: it is sensitive to initial conditions; it does not remove undesirable noise (e.g. variables that are not useful for

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

clustering); and it is optimal only for hyper-spherical clusters. In addition, its complexity in time is $O(nkl)$ and in space is $O(k)$, where n is the number of patterns, k is the number of clusters, and l the number of iterations. This complexity can be impractical for large datasets.

In this paper, we propose Discriminative Cluster Analysis (DCA) that alleviates some of the previously mentioned problems. DCA jointly optimizes for clustering and dimensionality reduction. In the first step, DCA finds a low dimensional projection of the data well suited for clustering by encouraging the preservation of distances between neighboring data points. Once the data is projected into a low dimensional space, DCA finds a "soft" clustering of the data. Later, this information is fed back into the dimensionality reduction step until convergence. Clustering in this subspace is less prone to local minima, it is faster to compute (especially for high dimensional data), and noisy dimensions that are not useful for clustering are removed. Moreover, it is often difficult to model correlations in high dimensional spaces, but by projecting them into a low dimensional space, these correlations can be modeled. Figure 1 shows the main benefits of DCA.

2. Previous work

This section reviews, in a unified matrix framework, previous work on k-means clustering, spectral methods and linear discriminant analysis, and points out the relationship between them.

2.1. K-means and spectral graph methods: a unified framework

K-means (MacQueen, 1967; Jain, 1988) is one of the simplest and most popular unsupervised learning algorithms to solve the clustering problem. Clustering refers to the partition of n data points into c disjoint clusters. k-means clustering splits a set of n objects into c groups by maximizing the between-clusters variation relative to within-cluster variation. In other words, k-means clustering finds the partition of the data that is a local optimum of the following energy function:

$$J(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_n) = \sum_{i=1}^{c} \sum_{j \in C_i} ||\mathbf{d}_j - \boldsymbol{\mu}_i||_2^2 \qquad (1)$$

where $\mathbf{d}_j$ (see notation [1]) is a vector representing the j^{th} data point and $\boldsymbol{\mu}_i$ is the geometric centroid of the data points for class i. The optimization criteria in eq. 1 can be rewritten in matrix form as:

$$E_1(\mathbf{M}, \mathbf{G}) = ||\mathbf{D} - \mathbf{M}\mathbf{G}^T||_F \qquad (2)$$
$$\textit{subject to } \mathbf{G}\mathbf{1}_c = \mathbf{1}_n \ \ \textit{and} \ \ g_{ij} \in \{0, 1\}$$

where $\mathbf{G} \in \Re^{n\times c}$ and $\mathbf{M} \in \Re^{d\times c}$. $\mathbf{G}$ is a dummy indicator matrix, such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0, 1\}$ and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class C_j, c denotes the number of classes and n is the number of samples. The columns of $\mathbf{D} \in \Re^{d\times n}$ contain the original data points, where d is the dimension of the data. The equivalence between the k-means error function eq. 1 and eq. 2 is only valid if $\mathbf{G}$ strictly satisfies the constraints.

The k-means algorithm performs coordinate descent in $E_1(\mathbf{M}, \mathbf{G})$. Given the actual value of the means $\mathbf{M}$, the first step finds for each data point $\mathbf{d}_j$, the $\mathbf{g}^j$ such that one of the columns is one and the rest 0 and minimizes eq. 2. The second step optimizes over $\mathbf{M} = \mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$, which is equivalent to compute the mean of each cluster. Although it can be proven that alternating these two steps will always terminate, the k-means algorithm does not necessarily find the optimal configuration over all possible assignments. The algorithm is significantly sensitive to the initial randomly selected clusters' centers; it typically runs multiple times, and the best solution is chosen. Despite these limitations, the algorithm is used frequently as a result of its ease of implementation and effectiveness.

Eliminating $\mathbf{M}$, eq. 2 can be rewritten as:

$$E_2(\mathbf{G}) = ||\mathbf{D} - \mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T||_F = tr(\mathbf{D}^T\mathbf{D})$$
$$-tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}) \geq \sum_{i=c+1}^{min(d,n)} \lambda_i \qquad (3)$$

where λ_i are the eigenvalues of $\mathbf{D}^T\mathbf{D}$. Minimizing eq. 3 is equivalent to maximizing $tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G})$. Ignoring the special structure of $\mathbf{G}$ and considering the continuous domain, the optimum $\mathbf{G}$ value that optimizes eq. 3 is given by the eigenvectors of the covariance matrix $\mathbf{D}^T\mathbf{D}$ and the error is $E_2 = \sum_{i=c+1}^{min(d,n)} \lambda_i$. (Ding & He, 2004; Zha et al., 2001) reported similar reasoning, they show that a lower bound of eq. 3 is given by the residual eigenvalues. The continuous solution of $\mathbf{G}$ lies in the $c-1$

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the j column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. d_{ji} is the i-th scalar element of the vector $\mathbf{d}^j$. All non-bold letters will represent variables of scalar nature. $diag$ is an operator that transforms a vector to a diagonal matrix or takes the diagonal of the matrix into a vector. $\circ$ denotes the Hadamard or point-wise product. $\mathbf{1}_k \in \Re^{k\times 1}$ is a vector of ones. $\mathbf{I}_k \in \Re^{k\times k}$ is the identity matrix. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$ and $|\mathbf{A}|$ denotes the determinant. $||\mathbf{A}||_F = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of a matrix.

subspace spanned by the first $c-1$ eigenvectors with highest eigenvalues (Ding & He, 2004) of $\mathbf{D}^T\mathbf{D}$.

Finally, it is worthwhile to point out the connections (Dhillon et al., 2004) between k-means and standard spectral graph algorithms, such as Normalized Cuts (Shi & Malik, 2000), by means of kernel methods. The kernel trick is a standard method for lifting the points of a dataset to a higher dimensional space, where points are more likely to be linearly separable (assuming that the correct mapping is found). Consider a lifting of the original points to a higher dimensional space, $\boldsymbol{\Gamma} = [\ \boldsymbol{\phi}(\mathbf{d}_1)\ \boldsymbol{\phi}(\mathbf{d}_2)\ \cdots\ \boldsymbol{\phi}(\mathbf{d}_n)\ ]$ where $\boldsymbol{\phi}$ is a high dimensional mapping. The kernelized version of eq. 2 will be:

$$E_3(\mathbf{M},\mathbf{G}) = ||(\boldsymbol{\Gamma} - \mathbf{M}\mathbf{G}^T)\mathbf{W}||_F \tag{4}$$

where a weighting matrix $\mathbf{W}$ for normalization purposes is introduced. Eliminating $\mathbf{M} = \boldsymbol{\Gamma}\mathbf{W}\mathbf{W}^T\mathbf{G}(\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{G})^{-1}$, it can be shown that:

$$E_3 \propto -tr((\mathbf{G}^T\mathbf{W}\mathbf{W}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{W}^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{W}\mathbf{W}^T\mathbf{G}) \tag{5}$$

where $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ is the standard affinity matrix in Normalized Cuts (Shi & Malik, 2000). After a change of variable $\mathbf{Z} = \mathbf{G}^T\mathbf{W}$, the previous equation can be expressed as $E_3(\mathbf{Z}) \propto -tr((\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{W}^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{W}\mathbf{Z}^T)$. Choosing $\mathbf{W} = diag(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{1}_n)^{-0.5}$ the problem is equivalent to solving the Normalized Cuts problem. This formulation is more general since it allows for arbitrary kernels and weights. In addition, the weight matrix can be used to reject the influence of pairs of data points with unknown similarity (i.e. missing data).

2.2. Linear Discriminant Analysis

The aim of most discriminant analysis methods (e.g. LDA) is to project the data into a space of lower dimension so that the classes are projected as far as possible from each other and the projection is compact within each cluster. LDA can be calculated by maximizing several optimization criteria (Fukunaga, 1990), and most of them are based on relations between the following covariance matrices, conveniently expressed in matrix form (de la Torre & Kanade, 2005):

$$\begin{aligned}
f\mathbf{S}_t &= \sum_{j=1}^{n}(\mathbf{d}_j - \mathbf{m})(\mathbf{d}_j - \mathbf{m})^T = \mathbf{D}\mathbf{P}_1\mathbf{D}^T \\
f\mathbf{S}_w &= \sum_{i=1}^{c}\sum_{\mathbf{d}_j \in C_i}(\mathbf{d}_j - \mathbf{m}_i)(\mathbf{d}_j - \mathbf{m}_i)^T = \mathbf{D}\mathbf{P}_2\mathbf{D}^T \\
f\mathbf{S}_b &= \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \mathbf{D}\mathbf{P}_3\mathbf{D}^T
\end{aligned}$$

where $f = n-1$, $\mathbf{P}_i$'s are projection matrices (i.e $\mathbf{P}_i^T = \mathbf{P}_i$ and $\mathbf{P}_i^2 = \mathbf{P}_i$) with the following expressions:

$$\begin{aligned}
\mathbf{P}_1 = \mathbf{I} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \quad \mathbf{P}_2 = \mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \\
\mathbf{P}_3 = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T
\end{aligned} \tag{6}$$

$\mathbf{S}_b$ is the between-class covariance matrix and represents the average of the distances between the mean of the classes. $\mathbf{S}_w$ represents the within-class covariance matrix and it is a measure of the average compactness of each class. Finally, $\mathbf{S}_t$ is the total covariance matrix. With the matrix expressions, it is straightforward to show that $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$. The upper bounds on the ranks of the matrices (if d¿¿n) are $c-1$, $n-c$, $n-1$ for $\mathbf{S}_b, \mathbf{S}_w, \mathbf{S}_t$ respectively. Note that eq. 2 is $tr(\mathbf{S}_w)$.

LDA computes a linear transformation of the data $\mathbf{B} \in \Re^{d \times k}$ that maximizes the distance between the means of the classes and minimizes the variance within clusters. The Rayleigh-like quotient is among the most popular LDA optimization criteria (Fukunaga, 1990), some are: $J_1(\mathbf{B}) = \frac{|\mathbf{B}^T\mathbf{S}_1\mathbf{B}|}{|\mathbf{B}^T\mathbf{S}_2\mathbf{B}|}$ $J_2(\mathbf{B}) = tr((\mathbf{B}^T\mathbf{S}_1\mathbf{B})(\mathbf{B}^T\mathbf{S}_2\mathbf{B})^{-1})$ $J_3(\mathbf{B}) = \frac{tr(\mathbf{B}^T\mathbf{S}_1\mathbf{B})}{tr(\mathbf{B}^T\mathbf{S}_2\mathbf{B})}$, where $\mathbf{S}_1 \in \{\mathbf{S}_b, \mathbf{S}_b, \mathbf{S}_t\}$ and $\mathbf{S}_2 \in \{\mathbf{S}_w, \mathbf{S}_t, \mathbf{S}_w\}$. A closed form solution to previous minimization problems is given by a generalized eigenvalue problem $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\Lambda$.

Previous Rayleigh quotient optimization procedures are not easy to modify in order to incorporate new constraints (e.g temporal constraints or geometric invariance). Formulating LDA as an error function will allow for a better understanding of the LDA limitations; moreover, further generalizations will be easier to formulate . Consider the following weighted between-class covariance matrix, $\hat{\mathbf{S}}_b = \mathbf{D}\mathbf{G}\mathbf{G}^T\mathbf{D}^T = \sum_{i=1}^{C}(\frac{n_i}{n})^2(\mathbf{m}_i)(\mathbf{m}_i)^T$, that favors the classes with more samples. $\mathbf{m}_i$ is the mean vector for the class i, and we assume the global mean (i.e. $\mathbf{m} = \frac{1}{n}\mathbf{D}\mathbf{1}_n$) is zero. In the line with earlier work on neural networks (Gallinari et al., 1991; Lowe & Webb, 1991), maximizing $J_4(\mathbf{B}) = tr((\mathbf{B}^T\hat{\mathbf{S}}_b\mathbf{B})(\mathbf{B}^T\mathbf{S}_t\mathbf{B})^{-1})$ is equivalent to minimizing:

$$E_4(\mathbf{B},\mathbf{V}) = ||\mathbf{G}^T - \mathbf{V}\mathbf{B}^T\mathbf{D}||_F \tag{7}$$

(Gallinari et al., 1991; Lowe & Webb, 1991) shows that:

$$E_4(\mathbf{B}) \propto -tr(((\mathbf{B}^T\mathbf{D}\mathbf{D}^T\mathbf{B})^{-1})\mathbf{B}^T\mathbf{D}\mathbf{G}\mathbf{G}^T\mathbf{D}^T\mathbf{B}) \tag{8}$$

This approach is appealing since (Baldi & Hornik, 1989) have shown that the surface of eq. 7 has a unique local minima , although it has several saddle points.

3. Discriminative Cluster Analysis

In the previous section, we have suggested a matrix formulation for a generative approach to understand the error function of k-means algorithm (unsupervised), and a matrix expression to derive an approximation of LDA (supervised). The aim of DCA is to combine clustering and dimensionality reduction in an unsupervised manner. In this section, we show how DCA finds a low dimensional projection that preserves the local structure of the data, improving clustering by iteratively computing $\mathbf{B}$ and $\mathbf{G}$.

3.1. Error function for LDA

The key aspect to simultaneously performing dimensionality reduction and clustering is the analysis of eq. 7. Ideally one would like to optimize eq. 7 w.r.t. $\mathbf{B}, \mathbf{G}$; however, directly optimizing eq. 7 has several drawbacks. First, eq. 7 biases the solution towards the classes that have more samples, because it maximizes $\hat{\mathbf{S}}_b = \mathbf{D}\mathbf{G}\mathbf{G}^T\mathbf{D}^T = \sum_{i=1}^{c}(\frac{n_i}{n})^2(\mathbf{m}_i)(\mathbf{m}_i)^T$. Secondly, eq. 7 does not encourage sparseness in $\mathbf{G}$ if $g_{ij} > 0$. Let $\mathbf{C} = \mathbf{B}^T\mathbf{D} \in \Re^{k\times n}$, eq. 7 is equivalent to $E_4 = tr(\mathbf{G}^T\mathbf{G}) - tr(\mathbf{G}^T\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}\mathbf{G})$. If g_{ij} are positive, minimizing $tr(\mathbf{G}^T\mathbf{G})$ does not encourage sparsitivity in $\mathbf{g}^i$ $\forall i$ ($\mathbf{g}^i$ represents the i row of $\mathbf{G}$, see notation).

In this section, we correct Eq. 7 to obtain the original LDA criteria by normalizing the error as follows:

$$E_5(\mathbf{B}, \mathbf{V}, \mathbf{G}) = ||(\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{V}\mathbf{B}^T\mathbf{D})||_F \qquad (9)$$

subject to the constraint that $g_{ij} \in \{0, 1\}$ and $\mathbf{G}\mathbf{1}_c = \mathbf{1}_n$. After eliminating $\mathbf{V}$, eq. 9 is equivalent to:

$$E_5(\mathbf{B}, \mathbf{G}) = ||(\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}\mathbf{G}^T(\mathbf{I}_n - \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C})||_F \quad (10)$$
$$\propto tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{B}(\mathbf{B}^T\mathbf{D}\mathbf{D}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{D}\mathbf{G}) \quad (11)$$

Equation eq. 11 can be re-written as $tr\big((\mathbf{B}^T\mathbf{D}\mathbf{D}^T\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{B})\big)$.
If $\mathbf{G}$ is known eq. 11 is the exact expression for LDA.

If the constraints on $\mathbf{G}$ are relaxed, an algorithm to minimize eq. 11 will alternate between solving the following eigenvalue problems:

$$\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{B} = \mathbf{D}\mathbf{D}^T\mathbf{B}\mathbf{\Lambda}_1 \qquad (12)$$
$$\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}\mathbf{G} = \mathbf{G}\mathbf{\Lambda}_2 \qquad (13)$$

Observe that in the continuous case there is an ambiguity because for any invertible matrix $\mathbf{T}_1 \in \mathbf{R}^{k\times k}$, $E_5(\mathbf{B}) = E_5(\mathbf{B}\mathbf{T}_1)$, similarly for any invertible matrix $\mathbf{T}_2 \in \mathbf{R}^{c\times c}$ $E_5(\mathbf{G}) = E_5(\mathbf{G}\mathbf{T}_2)$.

At this point, an interesting connection of the error function 11 with previous clustering techniques can be made. Let $\mathbf{B} = \mathbf{I}_d \in \Re^{d\times d}$ be the identity matrix (no projection is done). By assuming $\mathbf{G}$ is continuous, the low dimensional embedding for clustering will be computed as the eigenvectors of $\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}$. If $n >> d$ $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (i.e. SVD), where $\mathbf{U} \in \Re^{d\times d}$ $\mathbf{V} \in \Re^{n\times d}$ and $\mathbf{\Sigma} \in \Re^{d\times d}$, and assuming $\mathbf{D}\mathbf{D}^T$ is full rank, $\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D} = \mathbf{V}\mathbf{V}^T$, which is the affinity matrix used in (Costeira & Kanade, 1995) for motion segmentation problems .

3.2. Updating B

The optimal $\mathbf{B}$ given $\mathbf{G}$ can be computed in closed form by solving the following generalized eigenvalue problem (eq. 12) $\mathbf{D}\mathbf{R}\mathbf{D}^T\mathbf{B} = \mathbf{D}\mathbf{D}^T\mathbf{B}\mathbf{\Lambda}_1$, where $\mathbf{R} = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$. If $d << n$ and $\mathbf{D}\mathbf{D}^T$ is full rank, standard packages for generalized eigenvalue problems can be applied. However, for high dimensional data ($d >> n$) solving directly previous eigenvalue problem is not computationally efficient in either space or time. Fortunately, using the fact that the solutions of $\mathbf{B}$ are linear combinations of the data (i.e. $\mathbf{B}$=$\mathbf{D}\boldsymbol{\alpha}$), multiplying both sides by $\mathbf{D}^T$ and assuming $\mathbf{D}^T\mathbf{D}$ is invertible, the original eigenvalue problem is equivalent to solve $\mathbf{R}\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha} = \mathbf{D}^T\mathbf{D}\boldsymbol{\alpha}\mathbf{\Lambda}_1$, which is of much lower dimension ($n \times n$).

There are many ways of efficiently solving this generalized eigenvalue problem (e.g. subspace methods (de la Torre et al., 2005)). In this section, we explore closed form solutions that invert $\mathbf{D}^T\mathbf{D}$ and solve a regular eigenvalue problem (i.e. $(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{R}\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha} = \boldsymbol{\alpha}\mathbf{\Lambda}_1$). Assuming $\mathbf{D}^T\mathbf{D}$ is full rank, computing $(\mathbf{D}^T\mathbf{D})^{-1}$ can be a numerically unstable process, especially if $\mathbf{D}^T\mathbf{D}$ has some eigenvalues close to zero. A common approach to solving the ill-conditioning is to regularize the solution by factorizing $\mathbf{\Sigma} = \mathbf{D}^T\mathbf{D}$ as the sum of the outer products plus a scaled identity matrix, so that $\mathbf{\Sigma} \approx \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T + \sigma^2\mathbf{I}_d$. $\mathbf{V} \in \Re^{n\times k}$, $\mathbf{\Lambda} \in \Re^{k\times k}$ is a diagonal matrix. The parameters σ^2, $\mathbf{V}$, $\mathbf{\Lambda}$ are estimated by following a fitting approach that minimizes $E_c(\mathbf{V}, \mathbf{\Lambda}, \sigma^2) = ||\mathbf{\Sigma} - \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T - \sigma^2\mathbf{I}_n||_F$. After optimizing the parameters, (de la Torre & Kanade, 2005) have shown that: $\sigma^2 = tr(\mathbf{\Sigma} - \mathbf{V}\hat{\mathbf{\Lambda}}\mathbf{V}^T)/d - k$, $\mathbf{\Lambda} = \hat{\mathbf{\Lambda}} - \sigma^2\mathbf{I}_d$, where $\hat{\mathbf{\Lambda}}$ are the eigenvalues of the covariance matrix $\mathbf{\Sigma}$ and $\mathbf{V}$ the eigenvectors. After the factorization is found, the matrix inversion lemma (Golub & Loan, 1989) $((\mathbf{A}^{-1}+\mathbf{V}\mathbf{C}^{-1}\mathbf{V}^T)^{-1} = \mathbf{A} - (\mathbf{A}\mathbf{V}(\mathbf{C} + \mathbf{V}^T\mathbf{A}\mathbf{V})^{-1})\mathbf{V}^T\mathbf{A})$ is applied to invert $(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T + \sigma^2\mathbf{I}_d)^{-1}$ which results in:

$$(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T + \sigma^2\mathbf{I}_n)^{-1} = \frac{1}{\sigma^2}(\mathbf{I}_n - \frac{1}{\sigma^2}\mathbf{V}(\mathbf{\Lambda}^{-1} + \frac{\mathbf{I}_n}{\sigma^2})^{-1}\mathbf{V}^T)$$

Now solving $(\mathbf{I}_d - \frac{1}{\sigma^2}\mathbf{V}(\mathbf{\Lambda}^{-1} + \frac{\mathbf{I}_n}{\sigma^2})^{-1}\mathbf{V}^T)\mathbf{R}\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha} = \boldsymbol{\alpha}\mathbf{\Lambda}$ becomes more computationally tractable and numerically stable process.

The number of bases (k) are bounded by the number of classes, since $rank(\mathbf{DRD}^T) = c$. We typically choose $c-1$ to be consistent with LDA. Moreover, the best clustering results are achieved by projecting the data into a space of $c-1$ dimensions.

3.3. Updating G

If $\mathbf{A} = \mathbf{C}^T(\mathbf{CC}^T)^{-1}\mathbf{C} \in \Re^{n\times n}$, and $\mathbf{C} = \mathbf{B}^T\mathbf{D}$, eq. 11 is equivalent to:

$$E_5(\mathbf{G}) \propto tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{AG}) \qquad (14)$$

To impose non-negativity constraints in g_{ij}, we parameterize $\mathbf{G}$ as the product of two matrices $\mathbf{G} = \mathbf{V} \circ \mathbf{V}$ (Liu & Yi, 2003) and a gradient descent strategy is used to search for an optimum:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \eta\frac{\partial E_5(\mathbf{G})}{\partial \mathbf{V}} \qquad (15)$$
$$\frac{\partial E_5(\mathbf{G})}{\partial \mathbf{V}} = (\mathbf{I}_c - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)\mathbf{AG}(\mathbf{G}^T\mathbf{G})^{-1} \circ \mathbf{V}$$

The major problem with the update of eq. 15 is to determine the optimal η. In this case, η is determined with a line search strategy. To impose $\mathbf{G1}_c = \mathbf{1}_n$ in each iteration, the $\mathbf{V}$ is normalized to satisfy the constraint. Because eq. 15 is prone to local minima, this method starts from several random initial points and then selects the solution with minimum error.

This optimization problem is similar in spirit to recent work on clustering with non-negative matrix factorization (Zass & Shashua, 2005; Ding et al., 2005; Lee & Seung, 2000). However, we optimize a discriminative criteria rather than a generative one. On the other hand, we simultaneously compute the dimensionality reduction and clustering and a different optimization technique is used.

3.4. Initialization

At the beginning neither $\mathbf{G}$ nor $\mathbf{B}$ are known, but the matrix $\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$ can be estimated from data. Similarly to (He & Niyogi, 2003), we compute local information (i.e. $\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \in \Re^{n\times n}$) from data. We assume that $(\mathbf{G}^T\mathbf{G}) \approx s\mathbf{I}_c$, so that all the classes are equally distributed and s is the number of samples per class. $\mathbf{R} = \frac{1}{s}\mathbf{GG}^T$ is a hard-affinity matrix, where r_{ij} will be 1 if $\mathbf{d}_i$ and $\mathbf{d}_j$ are considered to be neighbors (i.e. belong to the same class). $\mathbf{R}$ can be estimated by simply computing the k nearest neighbors for each data point using Euclidian distance. To make $\mathbf{R}$ symmetric, the cases where $\mathbf{d}_i$ is within the k-neighborhood of $\mathbf{d}_j$ but not the opposite are filtered out. Figure 6.b shows an estimate of $\mathbf{R}$. In this example, 15 clusters (subjects), 10 samples per class and 9 nearest neighbors are selected. The samples are ordered by class. After factorizing (i.e. SVD) $\mathbf{R} = \mathbf{U\Sigma U}^T$, we normalize $\mathbf{R}$ as $\hat{\mathbf{R}} \approx \mathbf{U}_c\mathbf{U}_{\hat{c}}^T$, where $\mathbf{U}_c \in \mathbf{R}^{n\times c}$ are the first c eigenvectors of $\mathbf{R}$. $\hat{\mathbf{R}}$ will be the initial neighbor matrix.

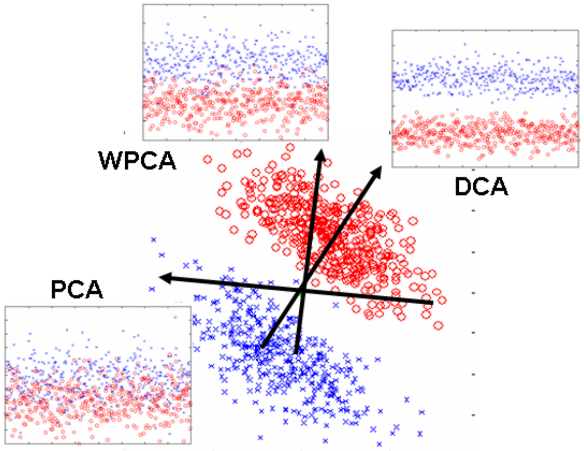

Figure 2. Two class toy problem. PCA, WPCA and DCA projections into 1 dimensional space.

3.5. Interpreting the weighted covariance matrix

A key aspect to interpret DCA is the understanding of the weighted covariance matrix $\mathbf{DRD}^T = \sum_{i=1}^n\sum_{j=1}^n r_{ij}\mathbf{d}_i\mathbf{d}_j^T$. Principal Component Analysis (PCA) (Jolliffe, 1986) computes a basis $\mathbf{B}$ that maximize the variance of the projected samples, i.e. PCA finds an orthonormal basis that maximizes $tr(\mathbf{B}^T\mathbf{DD}^T\mathbf{B}) = \sum_{i=1}^N ||\mathbf{B}^T\mathbf{d}_i||_2^2$. The PCA solution, $\mathbf{B}$, is given by the eigenvectors of $\mathbf{DD}^T$. Finding the leading eigenvectors of $\mathbf{DRD}^T$ is equivalent to maximize $tr(\mathbf{B}^T\mathbf{DRD}^T\mathbf{B}) = \sum_{i=1}^N\sum_{i=1}^N r_{ij}\mathbf{d}_j^T\mathbf{BB}^T\mathbf{d}_i$. If $\mathbf{R} = \mathbf{I}$ the standard PCA is the result; however, if $\mathbf{R}$ contains a block structure with the cluster information, the weighted covariance just maximizes the covariance within each cluster, finding a projection where the correlation between each pair of points within each cluster is maximized. Figure 2 shows a toy problem, where two Gaussian classes with an equal number of points are generated. The first eigenvector in PCA finds a direction of maximum variance that does not necessarily correspond to maximum discrimination (in fact, projecting the data into the first principal component the clusters overlap). If $\mathbf{R}$ is the initial matrix of neighbors, the first step of DCA finds a better projection for clustering.

Another possible unsupervised technique for dimensionality reduction preserving topological re-

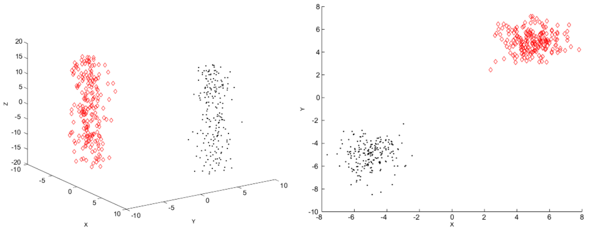

Figure 3. a) 2 classes of 3 dimensional data. b) Projection onto XY space.

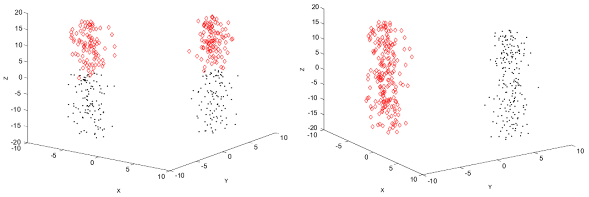

Figure 4. a) k-means clustering. b) DCA clustering.

lations is Weighted Principal Component Analysis. WPCA minimizes $E_5(\mathbf{B},\mathbf{C}) = ||(\mathbf{D} - \mathbf{BC})\mathbf{R}^{\frac{1}{2}}||_F$. After eliminating $\mathbf{C}$, minimizing the previous equation is equivalent to maximizing $E_5(\mathbf{B}) = tr((\mathbf{B}^T\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{DRD}^T\mathbf{B}))$. WPCA will be closely related to local preserving projection (He & Niyogi, 2003) with the right choice of weights. The optimum is achieved by solving the following eigenvalue problem $\mathbf{DRD}^T\mathbf{B} = \mathbf{B\Lambda}$, where $\mathbf{B}$ will be given by the eigenvectors of $\mathbf{DRD}^T$.

4. Experiments

4.1. Removing undesirable dimensions

The first experiment demonstrates the ability of DCA to deal with undesired dimensions not relevant for clustering. A toy problem is created as follows: 200 samples from a two-dimensional Gaussian distribution with mean $[-5,-5]$ and another 200 samples from another Gaussian with mean $[5,5]$ are generated (x and y dimensions). We add a third dimension generated with uniform noise between [0..35] (z dimension). Figure 3 shows the 200 samples of each class, in the original space (fig. 3.a) and the projection (fig. 3.b) onto x and y. k-means algorithm gets confused by the noise(fig. 4.a). Similarly, by projecting the data into the first two principal components, the wrong clustering is achieved, since PCA preserves the energy of the uniform noise which is not relevant for clustering. However, DCA (projecting into two dimensional space) is able to remove the noise and, it achieves the correct clustering, fig. 4.b. In this particular example 15 neighbors were initially selected and $\mathbf{B} \in \Re^{3\times 2}$.

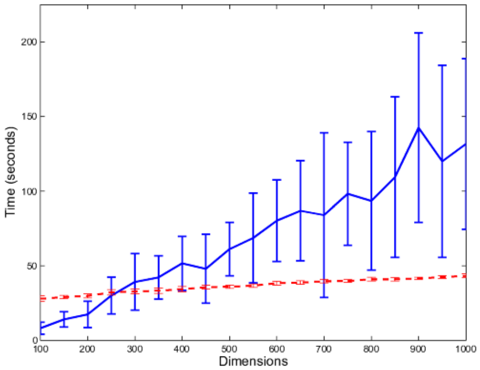

Figure 5. Time (seconds) versus number of dimensions. Blue straight line PCA+k-means, red dotted line DCA.

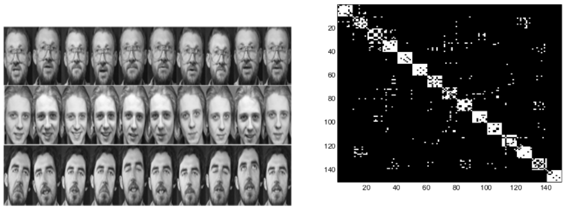

Figure 6. a) Some faces of the ORL data base. b) Estimate of $\mathbf{R}$ for 15 clusters (people), each cluster has 10 samples. The samples are ordered by clusters.

4.2. Computational efficiency

In this experiment, we have generated 400 samples of four x-dimensional Gaussians, with the dimension x ranging from 100 to 1000 in increments of 50. For two of the Gaussians, the means are 10 and -10 respectively; whereas, the other two have half of the dimensions as 10 and the other half -10, and vice versa. For each dimension and each Gaussian, we synthetically generate 400 samples and cluster by using k-means and DCA. Figure 5 shows the results of the time spent in clustering with k-means in the original spaces versus DCA. DCA is more computationally efficient as the number of samples increase.

4.3. Clustering faces

The final experiment shows results on clustering faces from the ORL face database (Samaria & Harter, 1994). The ORL face database is composed of 40 subjects and 10 images per subject. We randomly select k subjects from the database and add its 10 images to the data matrix $\mathbf{D} \in \Re^{d\times 10k}$ (e.g. fig. 6.a). Afterwards, we compute PCA, WPCA (with the initial matrix $\mathbf{R}$), PCA+LDA (preserving 95% of the energy in PCA) and DCA. After computing PCA, WPCA and PCA+LDA, we run the k-means (Matlab) on the projected samples 10 times and the best solution is chosen (the one with the least error). This procedure is repeated 40 times for different numbers of classes (between 4 and 40 subjects). To perform a fair comparison, we project the data into $classes - 1$ dimensions for all the methods. Fig. 7.a shows the PCA projection for the 10 classes case; fig. 7.b shows the DCA projection in the first step.

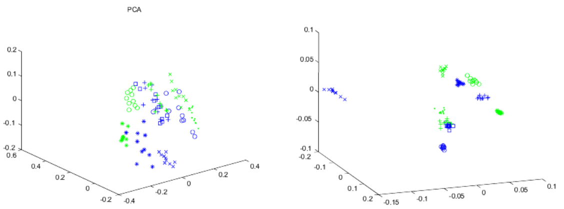

Figure 7. a) PCA projection. b)DCA projection.

To compute the accuracy of the results for a c cluster case, we compute a $c - by - c$ confusion matrix $\mathbf{C}$, where each entry c_{ij} is the number of faces in cluster i, which belong to class j. It is difficult to compute the accuracy by just using the confusion matrix $\mathbf{C}$ because it is not known which cluster matches which class. An optimal way to solve it (Zha et al., 2001; Knuth, 1993) is to compute the following maximization problem:

$$\max\ tr(\mathbf{CP}) \mid \mathbf{P}\ is\ a\ permutation\ matrix \qquad (16)$$

The accuracy is obtained by dividing the results for the number of data points to be clustered. To solve eq. 16, we use the classical Hungarian algorithm (Knuth, 1993). Table 1 shows the accuracy results (the mean and standard deviation over 40 runs) for different projection methods and different number of classes. DCA outperforms most of the methods when there are between 5 and 30 classes. For more classes $PCA + LDA$ performs marginally better.

C	PCA	WPCA	DCA	PCA+LDA
4	73±0%	1±0%	87±2%	**1**±0%
10	88±6%	95±6%	**97±4%**	88±8%
15	86±5%	88±4%	**96±1%**	82±6%
20	80±4%	84±4%	**87±2%**	83±4%
25	77±3%	80±4%	**87±2%**	80±4%
30	75±3%	79±3%	**81±3%**	81±4%
35	73±4%	77±3%	78±4%	**81±3%**
40	71±2%	74±3%	73±3%	**80±4%**

Table 1. Comparison of accuracy for several projection methods (same number of bases: classes-1).

Fig. 8 shows the accuracy in clustering for PCA+k-means vs. DCA. For a given number of clusters, we show the mean and variance of 40 realizations. DCA always outperforms PCA+k-means. In addition, the accuracy drops as the number of classes increases (as expected).

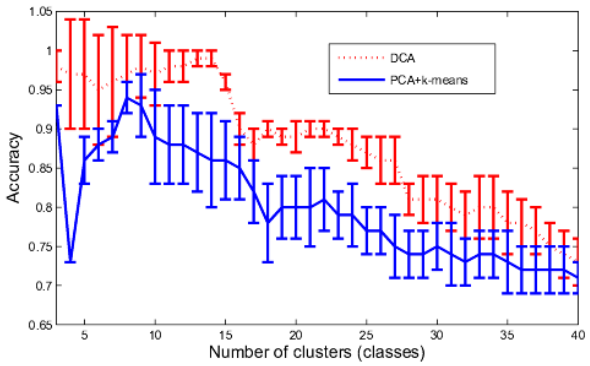

Figure 8. Accuracy of clustering versus the number of classes. Blue PCA and red DCA (dotted line).

5. Discussion and future work

In this paper we have proposed DCA, a new dimensionality reduction and clustering algorithm. DCA outperforms PCA+k-means, since it uses discriminative features for clustering rather than generative ones. Clustering in this space is less prone to local minima and removes irrelevant dimensions for clustering. Moreover, clustering in this low dimensional discriminative space is more computationally efficient than clustering in the original space. Additionally, we have constructed an error function for LDA.

However, several issues still need to be addressed. It still remains unclear how to select the optimal number of clusters; several model order selection (e.g. Minimum Description Length or Akaike information criterion) could be applied. On the other hand, DCA assumes that all the clusters have the same orientation (not necessarily spherical). Several extensions could be made in the case of non-Gaussian shape clusters, for instance using kernel methods. However, for huge amounts of high dimensional data ($d, n >> c$) the efficiency of the solution will be lost. Adding the constraint that $\mathbf{G}^T\mathbf{G} = \mathbf{I}$ will probably help the clustering process. An approximate approach to incorporating these constraints without transforming the problem into a quadratic programming (which is computationally expensive) is to add $\lambda tr(\mathbf{G}^T\mathbf{G})$ to eq. 11.

6. Acknowledgements

Thanks to the reviewers for their valuable comments and insights.

This work has been partially supported by MH R01 51435 from the National Institute of Mental Health, N000140010915 from the Naval Research Laboratory and by the Department of the Interior, National Business Center under contract no. NBCHD030010 and SRI International under subcontract no. 03-000211.

References

Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, *2*, 53–58.

Costeira, J., & Kanade, T. (1995). A multi-body factorization method for motion analysis. *International Conference on Computer Vision.*

de la Torre, F., Gross, R., Baker, S., & Kumar, V. (2005). Representational oriented component analysis (roca) for face recognition with one sample image per training class. *Computer Vision and Pattern Recognition.*

de la Torre, F., & Kanade, T. (2005). Multimodal oriented discriminant analysis. *International Conference on Machine Learning.*

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). A unified view of kernel k-means, spectral clustering and graph partitioning. *UTCS Technical Report TR-04-25.*

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. *International Conference on Machine Learning.*

Ding, C., He, X., & Simon, H. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. *Siam International Conference on Data Mining (SDM).*

Fukunaga, K. (1990). *Introduction to statistical pattern recognition, second edition.* Academic Press.Boston, MA.

Gallinari, P., Thiria, S., Badran, F., & Fogelman-Soulie, F. (1991). On the relations between discriminant analysis and multilayer perceptrons. *Neural Networks*, *4*, 349–360.

Golub, G., & Loan, C. F. V. (1989). *Matrix computations.* 2nd ed. The Johns Hopkins University Press.

He, X., & Niyogi, P. (2003). Locality preserving projections. *Neural Information Processing Systems.*

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: A review. *ACM Computing Surveys.*

Jain, A. K. (1988). *Algorithms for clustering data.* Prentice Hall.

Jolliffe, I. T. (1986). *Principal component analysis.* New York: Springer-Verlag.

Knuth, D. E. (1993). *The standford graphbase.* Addison-Wesley Publishing Company.

Lee, D., & Seung, H. (2000). Algorithms for non-negative matrix factorization. *Neural Information Processing Systems* (pp. 556–562).

Liu, W., & Yi, J. (2003). Existing and new algorithms for nonnegative matrix factorization. *University of Texas at Austin.*

Lowe, D. G., & Webb, A. (1991). Optimized feature extraction and the bayes decision in feed-forward classifier networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 355–364.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *5-th Berkeley Symposium on Mathematical Statistics and Probability.Berkeley, University of California Press.* (pp. 1:281–297).

Samaria, F., & Harter, A. (1994). Parameterization of a stochastic model for human face identification. *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision.*

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

Zass, R., & Shashua, A. (2005). A unifying approach to hard and probabilistic clustering. *International Conference on Computer Vision.*

Zha, H., Ding, C., Gu, M., He, X., & Simon., H. (2001). Spectral relaxation for k-means clustering. *Neural Information Processing Systems* (pp. 1057–1064).