Dimensionality Reduction for Spectral Clustering

Donglin Niu
Northeastern University
dniu@ece.neu.edu

Jennifer G. Dy
Northeastern University
jdy@ece.neu.edu

Michael I. Jordan
University of California, Berkeley
jordan@cs.berkeley.edu

Abstract

Spectral clustering is a flexible clustering methodology that is applicable to a variety of data types and has the particular virtue that it makes few assumptions on cluster shapes. It has become popular in a variety of application areas, particularly in computational vision and bioinformatics. The approach appears, however, to be particularly sensitive to irrelevant and noisy dimensions in the data. We thus introduce an approach that automatically learns the relevant dimensions and spectral clustering simultaneously. We pursue an augmented form of spectral clustering in which an explicit projection operator is incorporated in the relaxed optimization functional. We optimize this functional over both the projection and the spectral embedding. Experiments on simulated and real data show that this approach yields significant improvements in the performance of spectral clustering.

1 Introduction

Research in unsupervised learning has classically focused on two main kinds of data analysis problems—dimension reduction and clustering. Solutions to these problems are viewed as discovering statistical structure that is hoped to be useful for a wide range of subsequent analyses. But a useful statistical structure can have different definitions in different application domains. A more recent trend is to develop dimension reduction or clustering methods that directly aim at assisting in a specific downstream problem, such as classification or regression. This trend has classical antecedents (notably, linear discriminant analysis), and it is exemplified by highly-active areas such as sufficient dimension reduction [13] and semi-supervised learning [6].

The basic idea is that dimension reduction combats the curse of dimensionality, and success in this battle is readily measured by embedding the problem in a classification or regression setting. But in many application areas, unsupervised learning is often the end goal, even if it is often difficult to state such goals quantitatively. For example, the overall goal may be clustering for purposes of understanding or visualization. The curse of dimensionality is as serious an obstacle to this goal as it is to the goal of classification, and it is desirable to explore the use of dimension reduction in the service not of a downstream supervised learning problem but in the service of the unsupervised learning problem of clustering. While this general desideratum has been suggested before in various contexts [see, e.g., 11, 12, 10, 7], there has been comparatively little exploration of specific methods to date.

Our focus is the area of spectral clustering [17, 30] which uses graph cuts as objective functions for nonlinear data separation. Spectral clustering algorithms represent data as a graph where samples are vertices and edge weights represent the similarity between samples. Data are partitioned by finding a k-way graph cut in two steps: (1) find a *spectral embedding* by finding an eigenvector/eigenvalue decomposition of a Laplacian matrix; and (2) based on the embedding find a partition via a *rounding procedure*, which generally takes the form of a simplified clustering algorithm such as k-means. Spectral clustering has the virtue that it makes relatively weak assumptions regarding the shapes of clusters—clusters do not need to be convex or homogeneous. Moreover, it is applicable to a wide variety of data types and similarity functions. This flexibility, however, comes at a cost of lack of robustness; in particular, it has been observed that spectral clustering is quite sensitive to the presence of irrelevant and noisy dimensions in addition to signal-containing dimensions [2]. Of course, clustering in general is difficult in high-dimensional spaces; it is known, for example, that in high dimensions the distances between any two pairs of points are nearly constant for a wide variety of data distributions and distance functions [5]. Thus, it seems worthwhile to explore explicit strategies for finding the relevant low-dimensional subspace in which clustering structures reside, and we might expect that such strategies would be particularly beneficial

for spectral clustering.

Before spectral clustering is applied, one must first compute pair-wise similarities among data points. When some input features are irrelevant to the clustering task, they act as noise, distorting the similarities and confounding the performance of spectral clustering. Figure 1 row 2 shows an example on how irrelevant and noisy dimensions can mislead spectral clustering. The desired cluster embedding is a three ring structure in two relevant dimensions. Adding a third noisy dimension using a zero-mean Gaussian with variance σ_N and mixing the dimensions by a random projection V_{random}, $Data = Data * V_{random}$, we get a $3D$ scatter-plot as shown in subfigure (2a). Given the data in subfigure (2a) as the original input. Typical spectral clustering defines its similarity using all these dimensions. In subfigure (2c), we show the spectral similarity matrix utilized by spectral clustering. Because of the irrelevant and noisy dimensions, spectral clustering was not able to recover the three ring structure. Our goal in this paper is to learn the low-dimensional subspace that captures the relevant dimensions for defining the similarity graph to allow us to discover the underlying cluster structure.

In this paper, we introduce an approach that incorporates dimensionality reduction into spectral clustering to find the relevant low-dimensional subspace and clusters simultaneously. Another virtue of spectral clustering is that, it is based on an explicit optimization problem. The spectral embedding step is specified as the optimization of a tractable relaxation of the original intractable graph-partition problem. This provides us with a relatively straightforward way to incorporate dimension reduction into spectral clustering: We simply introduce a projection operator as an additional parameter in our problem and optimize the tractable optimization functional with respect to both the embedding and the projection operator. We do this by optimizing the embedding and the projection sequentially. Assuming a fixed projection, optimizing the embedding is simply an eigenproblem. Interestingly, as we show in Section 3, the optimization with respect to the projection and simultaneously learning the spectral cluster embedding has an interpretation as a solution to an unsupervised sufficient dimensionality reduction problem based on the Hilbert-Schmidt Independence Criterion (HSIC) [14].

There are several relevant threads of research in the literature. First, it is important to distinguish our approach from the common practice of using principal component analysis (PCA) as a preprocessing step before clustering [e.g., 29]. The directions of maximum variance of the data may have little relation to directions that reveal clustering, and our goal is precisely to use a clustering-related criterion to drive the choice of projection. Second, there are a variety of nonlinear dimension reduction methods—including kernel PCA [24], locally linear embedding (LLE) [23], Laplacian eigenmaps [4], and isometric feature mapping

(ISOMAP) [27]—that implicitly combine aspects of clustering with dimension reduction. Indeed, when using kernels based on radial basis functions, kernel PCA arguably can be viewed as an implicit clustering method. However, none of these nonlinear dimension reduction techniques perform selection and transformation in the original input feature space. Their assumption is that all of the original input features are relevant and they perform selection in a kernel space or embedding space. Our approach differs from these in that we learn the relevant low-dimensional subspace in the input space. This reflects our goal of reducing the sensitivity of spectral clustering to noisy input dimensions, and also has advantages for interpretability, which is often important in unsupervised learning. Note also that our framework is based on an explicit clustering criterion and an explicit dimension-reduction operator. Third, like graph fitting methods [8], we learn a similarity graph. But, their goal is to learn a graph that can serve as a general pre-processing step prior to classification, regression or clustering. In contrast, our work tries to learn a graph by learning a lower-dimensional subspace specifically for the purpose of clustering. Fourth, our work has relationships to semi-supervised metric learning [28], where a distance metric for clustering is learned, and to the work of [2], which focuses specifically on learning the weights for spectral clustering; however, these ideas make use of both labeled and unlabeled data, while our approach is entirely unsupervised. Finally, most closely related to our approach are LDA-k-means [10] and nonlinear adaptive distance metric learning (NAML) [7]. These algorithms perform data projection and clustering steps iteratively to enhance cluster quality until convergence. In LDA-k-means, both of these steps are carried out in the original space to optimize the k-means objective. The method thus inherits the disadvantages of k-means, notably the strong assumptions on cluster shapes. The NAML algorithm performs both the projection and clustering steps in kernel space, an idea reminiscent of kernel Fisher discriminant analysis (KFDA) [18]. Our method, on the other hand, performs spectral embedding in kernel space and data projection in the original space.

The remainder of this paper is organized as follows. In Section 2, we review spectral clustering. Section 3 presents sufficient dimensionality reduction for unsupervised learning and relates it to spectral clustering. In Section 4, we describe our dimensionality reduction for spectral clustering algorithm. Then, we present and discuss our experimental results on Section 5. Finally, we conclude in Section 6.

2   Background on Spectral Clustering

Spectral clustering can be presented from different points of view [17]; here, we focus on the graph partitioning viewpoint. We are given a set of n data samples, $\{x_1, \ldots, x_n\}$, with each x_i a column vector in R^d, and we are given a set of similarities, $\{k_{ij}\}$, between all pairs x_i and x_j,

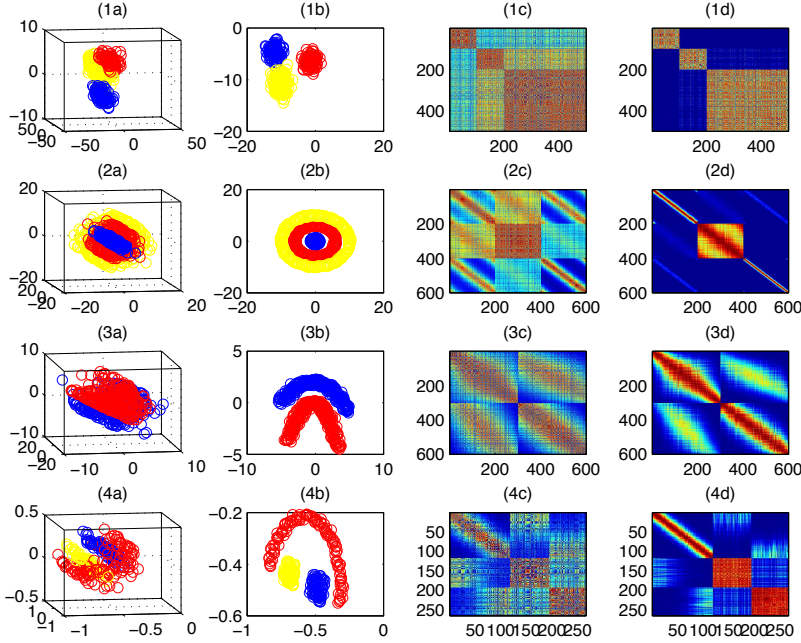**Donglin Niu, Jennifer G. Dy, Michael I. Jordan**

Figure 1: (1a), (2a), (3a) and (4a) show the scatter plots of the synthetic datasets 1, 2, 3 and 4 respectively in the original space. (1b), (2b), (3b) and (4b) show scatter plots of datasets 1, 2, 3 and 4 respectively in the reduced space discovered by our *DRSC* algorithm. (1c), (2c), (3c) and (4c) are the spectral (similarity) matrix of the data. (1d), (2d), (3d) and (4d) are the spectral (similarity) matrix in the learned reduced space.

where $k_{ij} \geq 0$. Let $G = \{V, E\}$ be a graph, with $V = \{v_1, \ldots, v_n\}$ the set of vertices and E the set of edges. Each vertex v_i in this graph represents a data sample x_i, with the similarities k_{ij} treated as edge weights. When there is no edge between v_i and v_j, $k_{ij} = 0$. Let us represent the similarity matrix as a matrix K with elements k_{ij}. This matrix is generally obtained from a kernel function, examples of which are the Gaussian kernel ($k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$) and the polynomial kernel ($k(x_i, x_j) = (x_i \cdot x_j + c)^p$).

The goal of spectral clustering is to partition the data $\{x_1, \ldots, x_n\}$ into k disjoint groups or partitions, $P_1, \ldots, P_k$, such that the similarity of the samples *between groups* is low, and the similarity of the samples *within groups* is high. There are several objective functions that capture this desideratum; in this paper we focus on the *normalized cut* objective. The k-way normalized cut, $Ncut(G)$, is defined as follows: $Ncut(P_1, ..., P_k) = \sum_{c=1}^{k} \frac{cut(P_c, V \setminus P_c)}{vol(P_c)}$, where the cut between sets $\mathcal{A}, \mathcal{B} \subseteq V$, $cut(\mathcal{A}, \mathcal{B})$, is defined as $cut(\mathcal{A}, \mathcal{B}) = \sum_{v_i \in \mathcal{A}, v_j \in \mathcal{B}} k_{ij}$, the *degree*, d_i, of a vertex, $v_i \in V$, is defined as $d_i = \sum_{j=1}^{n} k_{ij}$, the volume of set $\mathcal{A} \subseteq V$, $vol(\mathcal{A})$, is defined as $vol(\mathcal{A}) = \sum_{i \in \mathcal{A}} d_i$, and $V \setminus \mathcal{A}$ denotes the complement of $\mathcal{A}$. In this objective function, note that $cut(P_c, V \setminus P_c)$ measures the between cluster similarity and the within cluster similarity is captured by the normalizing term $vol(P_c)$. The next step is to rewrite $Ncut(G)$ using an indicator matrix U of cluster membership of size n by k and to note that

$Ncut(G)$ takes the form of a Rayleigh quotient in U. Relaxing the indicator matrix to allow its entries to take on any real value, we obtain a generalized eigenvector problem. That is, the problem reduces to the following relaxed $Ncut$ minimization:

$$\min_{U \in R^{n \times k}} \quad \text{trace}(U^T L U)$$
$$\text{s.t.} \quad U^T U = I. \tag{1}$$

where L is the normalized graph Laplacian, $L = I - D^{-1/2} K D^{-1/2}$, I is an identity matrix, D also called the degree matrix is a diagonal matrix whose diagonal entries are the degree d_i, and U is the spectral embedding matrix. Minimizing the relaxed $Ncut$ objective is equivalent to maximizing the relaxed normalized association $Nasso$ as follows:

$$\max_{U \in R^{n \times k}} \quad \text{trace}(U^T D^{-1/2} K D^{-1/2} U)$$
$$\text{s.t.} \quad U^T U = I. \tag{2}$$

From this point onwards, we refer to this maximization problem as our spectral clustering objective. The solution is to set U equal to the k eigenvectors corresponding to the largest k eigenvalues of the normalized similarity, $D^{-1/2} K D^{-1/2}$. This yields the spectral embedding. Based on this embedding, the discrete partitioning of the data is obtained from a "rounding" step. One specific rounding algorithm, due to [20], is based on renormalizing each row of U to have unit length and then applying k-means to the rows of the normalized matrix. We then assign each x_i to the cluster that the row u_i is assigned to.

3 Unsupervised Sufficient Dimensionality Reduction

Borrowing terminology from regression graphics [16, 15] and classical statistics, sufficient dimension reduction is dimension reduction without loss of information. Sufficient dimensionality reduction [16, 15, 21, 13] aims at finding a linear subspace $\mathcal{S} \subset \mathcal{X}$ such that $\mathcal{S}$ contains as much predictive information regarding the output variable Y as the original space $\mathcal{X}$. We can express this in terms of conditional independence as follows:

$$Y \perp\!\!\!\perp X | W^T X \qquad (3)$$

where W is the orthogonal projection of $\mathcal{X}$ onto subspace $\mathcal{S}(W)$ and $\perp\!\!\!\perp$ denotes statistical independence. The subspace $\mathcal{S}(W)$ is called a dimension reduction subspace. This statement equivalently says that the conditional distribution of $Y|X$ is the same as $Y|W^T X$, which implies that replacing X with $W^T X$ will not lose any predictive information on Y. There are many such subspaces because if $\mathcal{S}(W_1)$ is a dimension reduction subspace, any subspace $\mathcal{S}$ which contains subspace $\mathcal{S}(W_1)$, $\mathcal{S}(W_1) \subset \mathcal{S}$, will also be a dimension reduction subspace. Note too that a dimension reduction subspace always exists with W equal to the identity matrix I serving as a trivial solution. The intersection of all such subspaces or the smallest dimension reduction subspace is called the *central subspace*.

The literature on sufficient dimensionality reduction has focused on the supervised setting [16, 15, 21, 13]. This paper addresses finding the central subspace in the unsupervised setting, and in particular for clustering. In the supervised case, they have Y to guide the search for the central subspace. In the unsupervised case, Y is unknown and must be learned. To learn Y, we rely on criterion functions; in our case, we utilize the spectral clustering criterion where we estimate Y by U in Equation 2.

Recently, kernel measures have been utilized to find the central subspace [21, 13]. To perform sufficient dimensionality reduction, Equation 3, some way of measuring the independence/dependence between X and Y is needed. Mutual information is an example of a criterion for measuring dependence, however it requires estimating the joint distribution between X and Y. The work by [13] and [14] provide a way to measure dependence among random variables without explicitly estimating joint distributions. The basic idea is to map random variables into reproducing kernel Hilbert spaces (RKHSs) such that second-order statistics in the RKHS capture higher-order dependencies in the original space. One such measure is the Hilbert-Schmidt Independence Criterion (HSIC) [14]. HSIC is the Hilbert-Schmidt norm of the cross-covariance operator on two random variables. Interestingly, the spectral clustering objective, Equation 2, can be expressed in terms of the HSIC measure. This relationship is also noted in [25]. The empirical approximation to $\text{HSIC}(X, U)$ is:

$$\text{HSIC}(X, U) = (n - 1)^{-2} \text{trace}(K_1 H K_2 H),$$

where K_1, $K_2 \in R^{n \times n}$ are the Kernel gram matrices $K_{1,ij} = k_1(x_i, x_j)$, $K_{2,ij} = k_2(u_i, u_j)$ and H is a centering matrix. For notational convenience, let us assume that K_1 and K_2 are centered and ignore the scaling factor $(n - 1)^{-2}$, and use $\text{HSIC}(X, U) = \text{trace}(K_1 K_2)$. Let $K_1 = D^{-1/2} K D^{-1/2}$, where K is the similarity kernel with elements, $k_{ij} = k(x_i, x_j)$, and $K_2 = UU^T$. Then,

$$\begin{aligned}
\text{HSIC}(X, U) &= \text{trace}(D^{-1/2} K D^{-1/2} UU^T) \\
&= \text{trace}(U^T D^{-1/2} K D^{-1/2} U),
\end{aligned}$$

which is the spectral clustering objective.

Assuming the labels U are known, we can estimate the central subspace by optimizing for W that maximizes the $\text{HSIC}(W^T X, U)$ dependence between $W^T X$ and U, where $k_{ij} = k(W^T x_i, W^T x_j)$ and $W^T W = I$. We can thus perform sufficient dimensionality reduction in the unsupervised setting by finding the central subspace and U that simultaneously maximize $\text{trace}(U^T D^{-1/2} K D^{-1/2} U)$. We describe this approach in detail in the next section.

4 Dimension Reduction for Spectral Clustering

In spectral clustering, the kernel similarity is defined on all the features. However, some features or directions may be noisy or irrelevant. Our goal is to project data onto a linear subspace and subsequently perform spectral clustering on the projected data. Moreover, we wish to couple these steps so that the projection chosen is an effective one for clustering as measured by the normalized association criterion. We achieve this goal by introducing a projection operator into the spectral clustering objective. Specifically, in computing for the similarity matrix K, we first project to a low-dimensional subspace by calculating $k(W^T x_i, W^T x_j)$, where $W \in R^{d \times q}$ is a matrix that transforms $x_i \in R^d$ in the original space to a lower dimensional space q ($q < d$). For example, if using a Gaussian kernel, the kernel function is defined as

$$k(W^T x_i, W^T x_j) = \exp(- \left\| W^T x_i - W^T x_j \right\|^2 / 2\sigma^2) \qquad (4)$$

For identifiability reasons, we constrain W to be orthonormal: $W^T W = I$. We then formulate the spectral clustering objective on the low-dimensional subspace as follows:

$$\begin{aligned}
\max_{U \in R^{n \times k}, W \in R^{d \times q}} \quad & \text{trace}(U^T D^{-1/2} K D^{-1/2} U) \\
\text{s.t.} \quad & U^T U = I \\
& k_{ij} = k(W^T x_i, W^T x_j), \text{ i,j=1}, \ldots, \text{n} \\
& W^T W = I,
\end{aligned} \qquad (5)$$

where K has elements $k_{ij} = k(W^T x_i, W^T x_j)$ and where D is the degree matrix $d_{ii} = \sum_j k(W^T x_i, W^T x_j)$.

We optimize this objective function using a coordinate ascent algorithm:

1. **Assuming W is fixed, optimize for U.** With the projection operator W fixed, we compute the similarity and degree matrices, K and D, respectively. We set U equal to the first k eigenvectors (corresponding to the largest k eigenvalues) of $D^{-1/2} K D^{-1/2}$.

2. **Assuming U is fixed, optimize for W.** With U fixed, each row of U is the spectral embedding of each data instance. We utilize a dimension growth algorithm to optimize the objective with W. First, we set the dimensionality of the subspace to be one, w_1. We use gradient ascent to optimize w_1, where w_1 is initialized by random projection and normalized to have norm 1. We, then, increase the dimensionality by one and optimize for w_2. w_2 is initialized by random projection, then projected to the space orthogonal to w_1, and finally normalized to have norm 1. We decompose the gradient of w_2 into two parts,

$$\nabla f = \nabla f_{proj} + \nabla f_{\perp} \qquad (6)$$

∇f_{proj} is the projection of ∇f to the space spanned by w_1 and w_2, and $\nabla f_{\perp}$ is the component orthogonal to ∇f_{proj} ($\nabla f_{proj} \perp \nabla f_{\perp}$). $\nabla f_{\perp}$ is normalized to have norm 1. We update w_2 according to the following equation

$$w_{2,new} = \sqrt{1 - \gamma^2} w_{2,old} + \gamma \nabla f_{\perp} \qquad (7)$$

The step size γ is set by line search satisfying the two Wolfe conditions. Repeat Equation 7 up to convergence. Because w_1 and w_2 are initially set to be orthonormal and w_2 is updated according to the above equation, w_2 and w_1 will remain orthonormal. w_j is optimized in the same way. w_j is updated orthogonal to $w_1, w_2, \ldots, w_{j-1}$. Once we have the desired number of dimensions q, we repeat Equation 7 for each w_j, $j = 1, \ldots, q$ until convergence.

We repeat these two steps iteratively until convergence. After convergence, we obtain the discrete clustering by using k-means in the embedding space U. Algorithm 1 provides a summary of our approach, we call Dimension Reduced Spectral Clustering (*DRSC*).

Applying *DRSC* to Gaussian and Polynomial Kernels. More specifically, we provide here details on how to implement *DRSC* to two widely used kernels: Gaussian and polynomial kernels. Different kernels only vary Step 2 of *DRSC*.

Algorithm 1 Dimension Reduced Spectral Clustering (*DRSC*)

Input: Data x_i, number of clusters k.
Initialize: Set $W = I$ (i.e., use the original input data).
Step 1: Given W, find U.
Calculate kernel similarity matrix K and normalized similarity matrix $D^{-1/2} K D^{-1/2}$. Keep the eigenvectors with the largest k eigenvalues of the normalized similarity $D^{-1/2} K D^{-1/2}$ to form U.
Step 2: Given U, find W.
Optimize W by dimension growth algorithm. Project data into subspace formed by W.
REPEAT steps 1 and 2 until convergence of the $Nasso$ value.
k**-means step:** Form n samples $y_i \in R^k$ from the rows of U. Cluster the points y_i, $i = 1, \ldots, n$, using k-means into k partitions, $P_1, \ldots, P_k$.
Output: Partitions $P_1, \ldots, P_k$ and the transformation matrix W.

Gaussian Kernel Case: For Step 2, we assume U is fixed, optimize for W. With the Gaussian kernel, we can re-write the objective function as follows:

$$\max_W \quad \sum_{ij} \frac{u_i^T u_j}{d_i d_j} \exp\left(-\frac{\Delta x_{ij}^T W W^T \Delta x_{ij}}{2\sigma^2}\right) \qquad (8)$$
$$\text{s.t.} \quad W^T W = I$$

where Δx_{ij} is the vector $x_i - x_j$, and $\Delta x_{ij}^T W W^T \Delta x_{ij}$ is the l^2 norm in subspace W. The above objective can be expressed as

$$\max_W \quad \sum_{ij} \frac{u_i^T u_j}{d_i d_j} \exp\left(-\frac{\text{trace}(W^T \Delta x_{ij} \Delta x_{ij}^T W)}{2\sigma^2}\right) \qquad (9)$$
$$\text{s.t.} \quad W^T W = I$$

or

$$\max_W \quad \sum_{ij} \frac{u_i^T u_j}{d_i d_j} \exp\left(-\frac{w_1^T A_{ij} w_1 + w_2^T A_{ij} w_2 + \ldots}{2\sigma^2}\right)$$
$$\text{s.t.} \quad W^T W = I$$
$$(10)$$

where w_i is the ith column of W, and A_{ij} is the d by d semidefinite positive matrix $\Delta x_{ij} \Delta x_{ij}^T$. In this step, we assume $\frac{u_i^T u_j}{d_i d_j}$ is fixed. Note that $w^T A w$ is a convex function. Thus, the summation of $w_i^T A w_i$ is convex. $exp(-y)$ is a decreasing function, so $\exp\left(-\frac{w_1^T A w_1 + w_2^T A w_2 + \ldots}{2\sigma^2}\right)$ is a concave function. Each component w_i must be orthogonal to each other to form a subspace. Since W with this constraint is not a convex set, the optimization problem is not a convex optimization. We then apply the dimension growth algorithm described earlier. Using the property of the exponential function, the objective becomes:

$$\max_W \sum_{ij} \frac{u_i^T u_j}{d_i d_j} \exp\left(-\frac{w_1^T A_{ij} w_1}{2\sigma^2}\right) \exp\left(-\frac{w_2^T A_{ij} w_2}{2\sigma^2}\right)$$
$$(11)$$

With w_1 fixed, the partial derivative with respect to w_2 is:

$$\sum_{ij} -\frac{u_i^T u_j}{d_i d_j} \frac{1}{\sigma^2} g(w_1) \exp(-\frac{w_2^T A_{ij} w_2}{2\sigma^2}) A_{ij} w_2 \quad (12)$$

where $g(w_1)$ is $\exp(-\frac{w_1^T A_{ij} w_1}{2\sigma^2})$. We update w_2 and w_j by Eqn. 7.

Polynomial Kernel Case: For the polynomial kernel, the kernel similarity in the projected subspace is $k(W^T x_i, W^T x_j) = (x_i^T W W^T x_j + c)^p$. The derivative of this kernel function with respect to W is

$$\frac{\partial k_{ij}}{\partial W} = p(x_i^T W W^T x_j + c)^{p-1}(x_j x_i^T + x_j x_i^T) W \quad (13)$$

W can be optimized by the modified gradient ascent algorithm in a similar way as that of the Gaussian case in Step 2 of *DRSC* but using this polynomial gradient equation.

Remark: The dimension growth algorithm will converge to a local optimum.

In the algorithm, we use update Eqn. 7, with $\gamma > 0$ satisfying the two Wolfe conditions. $\langle \nabla f_\perp, \nabla f'(w) \rangle = \langle \nabla f_\perp, \nabla f_\perp + \nabla f_{proj} \rangle = \langle \nabla f_\perp, \nabla f_\perp \rangle \geq 0$, thus $\nabla f_\perp$ is an ascent direction (i.e., it gives $f(w_{new}) > f(w_{old})$). $\langle, \rangle$ is the inner product operator. The algorithm will generate a sequence of w with $f(w_n) > f(w_{n-1}) > f(w_{n-2}) \dots$. The objective function is upper bounded in both steps. In Step 1, the objective is bounded by k if using k eigenvectors. In Step 2, if each element in the kernel similarity matrix is bounded, the objective is bounded. For the Gaussian kernel, $exp(-\frac{w^T A w}{2\sigma^2}) < 1$. For the polynomial kernel, using Cauchy inequality, $(x_i^T W W^T x_j + c)^p \leq (|x_i^T W W^T x_j| + c)^p \leq (|W^T x_i||W^T x_j| + c)^p \leq (|x_i||x_j| + c)^p$. This kernel is then bounded for finite and positive c and p if each original input x_i are finite. Assuming these conditions are held, the algorithm will converge to a local optimum.

Initialization. Our approach is dependent on initial parameter values. We can simply start by setting the kernel similarity using all the features $W_{(0)} = I$. Then calculate the embedding U using all the features. If the data has many possible clustering interpretations as explored in [22], this initialization will lead to the solution with the strongest clustering structure.

Computational Complexity. Calculating the similarity matrix K can be time consuming. We apply incomplete Cholesky decomposition as suggested in [1] giving us an approximate similarity matrix $\widetilde{K}$. The complexity of calculating this matrix is $O(ns^2)$, where n is the number of data points, s is the size of the Cholesky factor $\widetilde{G}$, where $\widetilde{K} = \widetilde{G}\widetilde{G}^T$. We set s such that the approximate error is less than $\epsilon = 10^{-4}$. Thus, the complexities of the

eigen-decomposition and derivative computations are now $O(ns^2)$ and $O(ns)$. The complexity of the overall algorithm is $O((ns^2 + nsrd)t)$, where d is the reduced dimensionality, r is the number of steps in the gradient ascent in Step 2, and t is the number of overall iterations.

5 Experiments

In this section, we present an empirical evaluation of our *DRSC* algorithm on both synthetic and real data. We compare *DRSC* against standard k-means (*k-means*), standard spectral clustering (*SC*) [20], PCA followed by spectral clustering (*PCA+SC*), adaptive Linear Discriminant Analysis combined with k-means (*LDA-k-means*) [10] and weighted kernel k-means combined with kernel Fisher discriminative analysis (*WKK-KFD*) [18]. Standard k-means and spectral clustering serve as our baseline algorithms. *PCA+SC* applies a dimensionality reduction algorithm, principal component analysis (PCA) in particular, before applying spectral clustering. *LDA-k-means* iteratively applies k-means and LDA until convergence, where the initialization is a step of PCA followed by k-means. In addition, we compare our algorithm to a kernel version of *LDA-k-means*, where we combine kernel k-means and kernel LDA (*WKK-KFD*). This method iteratively applies weighted kernel k-means and kernel Fisher discriminative analysis until convergence. As pointed out in [9], weighted kernel k-means is equivalent to spectral clustering, if we set the weight for each data instance according to its degree in the Laplacian matrix. We employ a Gaussian kernel for all spectral/kernel-based methods and set the kernel width by 10-fold cross-validation using the mean-squared error measure from the k-means step, searching for width values ranging from the minimum pairwise distance to the maximum distance of points in the data. For all methods running k-means, we initialize k-means with 10 random re-starts and select the solution with the smallest sum-squared error. We set the convergence threshold $\varepsilon = 10^{-4}$ in all experiments. To be consistent with LDA, we reduce the dimensionality for all methods to $k - 1$, where k is the number of clusters. In our dimension growth algorithm, at each time we add an extra dimension, if the normalized association value does not increase, we can stop the dimension growth. However, to be consistent and fair with the other methods, we simply use $k - 1$. Determining the cluster number is not trivial and remains an open research problem in spectral clustering. One possible way of selecting k is by checking the eigen-gap. Again, to be consistent and for ease of comparison for all methods, we assume it is known and we set it equal to the number of class labels for all methods.

The evaluation of clustering algorithms is a thorny problem. However, a commonly accepted practice in the community is to compare the results with known labeling. We measure the performance of our clustering methods based on the normalized mutual information (NMI) [26] be-
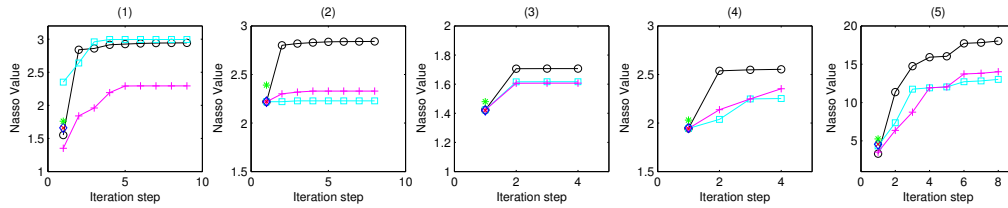
Figure 2: (1), (2), (3) and (4) are the $Nasso$ values of synthetic datasets 1, 2, 3 and 4 respectively, and (5) is for the real face data obtained by our *DRSC* algorithm (black line with circles), *LDA-k-means* (cyan line with square) and *WKK-KFD* (magenta line with plus) in each iteration. k-means (red cross), *SC* (blue diamond) and *PCA + SC* (green asterisk) results are also shown.
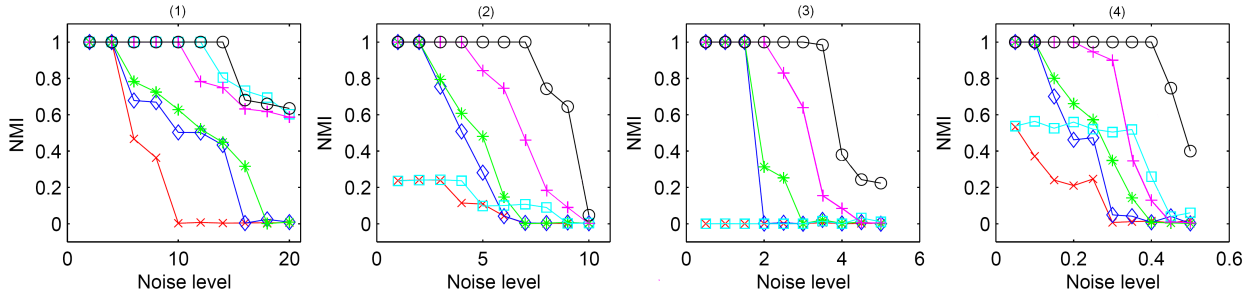

Figure 3: NMI values discovered by different clustering algorithms as a function of increasing noise levels for synthetic datasets 1, 2, 3 and 4. Red lines with crosses are results for the *k-means* algorithm. Blue lines with diamonds are results for spectral clustering. Cyan Lines with squares are results for *LDA-k-means*. Green lines with asterisks are results of *PCA+SC*. Magenta lines with pluses are results for *WKK-KFD*, and black lines with circles are results for *DRSC*.

tween the clusters found by these methods with the "true" class labels. We normalize to the range $[0, 1]$ by defining $NMI(X,Y) = MI(X,Y)/\sqrt{H(X)H(Y)}$, where $MI(X,Y)$ denotes the mutual information and where $H(X)$ and $H(Y)$ denote the entropy of X and Y. Higher NMI values mean higher agreement with the class labels.

5.1 Synthetic Data

To get a better understanding of our method, we first perform experiments on synthetic data. Synthetic datasets were generated in the following way. First, we embedded linear or nonlinear structures involving clusters, rings or hooks in two dimensions. The third dimension was a noise dimension, drawn from $N(0, \sigma_N^2)$. We performed random projection, $Data = Data * V_{random}$, to mix these dimensions, where V_{random} is a random orthonormal transformation matrix, $V_{random}^T V_{random} = I$. Data 1 has three Gaussian clusters in the two relevant dimensions. Data 2 and 3 are based on three rings and two hooks. Data 4 is a mixture of linear and nonlinear structures with two compact clusters and one hook.

Figure 1 shows the data in the original feature space (a) and in the reduced space discovered by our algorithm (b). The figure also shows the spectral (similarity) matrix of the data in the original space (c) and in the reduced space (d). From Figure 1 (a) and (c), we see that while the data and their spectral (similarity) matrices in the original

space have some structure, this structure is overwhelmed by noise. Indeed, at this noise level ($\sigma_N = 7, 5, 3.5, 0.2$ respectively), spectral clustering cannot find the correct partitions. On the other hand, from Figure 1 (b) and (d), we see that the data and the spectral (similarity) matrix in the reduced space discovered by *DRSC* show strong cluster structures. Moreover, spectral clustering in the reduced space can discover the underlying partitions. Figure 2 (1-4) displays the $Nasso$ value obtained by *DRSC* and the other methods as a function of iteration. Non-iterative methods are shown as just a point at iteration 1. The figure confirms that *DRSC* increases the $Nasso$ value in each step. Moreover, *DRSC* obtained the highest $Nasso$ score compared to competing methods at convergence. Since synthetic data 1 is linearly separable, *LDA-k-means* performed as well as *DRSC* on this dataset, but performed poorly on the other data sets.

In Figure 3, we show a comparison of the different methods in terms of NMI as the noise level σ^2 is varied. Note that our proposed *DRSC* method is the most robust to noise. *LDA-k-means* is satisfactory for Synthetic Data 1 where the clusters are spherical, but fails for the arbitrary-shaped data. In the presence of noise, *k-means* fails even for spherical clusters (Data 1). Because *WKK-KFD* can capture clusters with arbitrary shapes, it performed better than *LDA-k-means* on Data 2, 3 and 4. It is better than spectral clustering but is much worse than *DRSC*. This is because *WKK-KFD* only reduces the dimension in the em-

Table 1: NMI for Real Data

	FACE	MACHINE SOUND	HD DIGITS	CHART	GLASS	SATELLITE
k-means	0.71 ± 0.03	0.61 ± 0.03	0.60 ± 0.03	0.66 ± 0.01	0.43 ± 0.03	0.41 ± 0.02
LDA-k-means	0.76 ± 0.03	0.71 ± 0.03	0.67 ± 0.02	0.75 ± 0.02	0.32 ± 0.02	0.42 ± 0.03
SC	0.75 ± 0.02	0.79 ± 0.02	0.65 ± 0.02	0.68 ± 0.02	0.36 ± 0.02	0.40 ± 0.02
PCA+SC	0.75 ± 0.02	0.81 ± 0.03	0.73 ± 0.02	0.69 ± 0.03	0.33 ± 0.02	0.42 ± 0.02
WKK-KFD	0.81 ± 0.03	0.80 ± 0.02	0.69 ± 0.03	0.69 ± 0.02	0.32 ± 0.03	0.43 ± 0.03
DRSC	$\mathbf{0.87 \pm 0.03}$	$\mathbf{0.85 \pm 0.02}$	$\mathbf{0.79 \pm 0.02}$	$\mathbf{0.78 \pm 0.02}$	$\mathbf{0.45 \pm 0.02}$	$\mathbf{0.46 \pm 0.03}$

bedding space, whereas our *DRSC* approach reduces the subspace dimension in the input space. *PCA+SC* does not help spectral clustering much in dealing with noise. Spectral and *k-means* clustering performed poorly in the presence of noise; notice that when the noise is large, the NMI values drop rapidly.

5.2 Real Data

We now test on real data to investigate the performance of our algorithm. In particular, we test on face images, machine sounds, digit images, chart, glass data and satellite data. The face dataset from the UCI KDD archive [3] consists of 640 face images of 20 people taken at varying poses (straight, left, right, up), expressions (neutral, happy, sad, angry), eyes (wearing sunglasses or not). Note that identity is the dominant clustering structure in the data compared to pose, expression and eyes. The machine sound data is a collection of acoustic signals from accelerometers. The goal is to classify the sounds into different basic machine types: `pump`, `fan`, `motor`. We represent each sound signal by its FFT (Fast Fourier Transform) coefficients, providing us with $100,000$ coefficients. We select the 1000 highest values in the frequency domain as our features. The multiple digit feature dataset [19] consists of features of hand-written digits ('0'–'9') extracted from a collection of Dutch utility maps. Patterns have been digitized in binary images. These digits are represented by several feature subsets. In the experiment, we use the profile correlation feature subset which contains 216 features for each instance. The chart dataset [19] contains 600 instances each with 60 features of six different classes of control charts. The glass dataset [19] contains 214 instances with 10 features. One feature is the refractive index and nine features describe the chemical composition of glass. The satellite dataset [19] consists of 7 kinds of land surfaces. Features are multi-spectral values of pixels in 3×3 neighborhoods in a satellite image.

From Table 1, we observe that compared to competing algorithms, our *DRSC* algorithm obtained the best clustering results in terms of NMI (where the best values are shown in bold font). Similar to the results on synthetic data, we observe that *LDA-k-means* in general improves the performance of *k-means*. *PCA+SC* performs similarly or slightly better than spectral clustering, *SC*. *WKK-KFD*

is better than *SC*, but *DRSC* performs the best in all cases. *DRSC* led to better results than *WKK-KFD*, because *WKK-KFD* only reduces the dimension in the embedding space, whereas our *DRSC* approach reduces the subspace dimension in the input space. We take a closer look at the results for the face data. We observe that spectral clustering discovers a reasonable clustering based on identities. However, the spectral clustering results show interference from the pose aspect of the data. Our algorithm, on the other hand, focuses solely on the identity, which is the strongest clustering structure in the data. In Figure 2 (5), we show $Nasso$ values obtained by different algorithms with respect to the number of iterations for the face data. The plot confirms that our approach increases $Nasso$ in each step. In addition, the $Nasso$ value at convergence is close to the number of different persons (20) and the $Nasso$ value reached is the highest compared to competing methods.

6 Conclusions

Dimension reduction methods are increasingly being refined so as to find subspaces that are useful in the solution of downstream learning problems, typically classification and regression. In this paper we have presented a contribution to this line of research in which the downstream target is itself an unsupervised learning algorithm, specifically spectral clustering. We have focused on spectral clustering due to its flexibility, its increasing popularity in applications and its particular sensitivity to noisy dimensions in the data. We have developed a dimension reduction technique for spectral clustering that incorporates a linear projection operator in the relaxed optimization functional. We have shown how to perform optimization in this functional in the spectral embedding and the linear projection. Our results on synthetic and real data show that our approach improves the performance of spectral clustering, making it more robust to noise.

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Re-*

search, 3:1–48, 2002.

[2] F. R. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

[3] S. D. Bay. The UCI KDD archive, 1999.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In *International Conference on Database Theory*, pages 217–235, 1999.

[6] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[7] J. H. Chen, Z. Zhao, J. P. Ye, and H. Liu. Nonlinear adaptive distance metric learning for clustering. In *ACM SIGKDD Intn'l Conference on Knowledge Discovery and Data Mining*, pages 123–132, 2007.

[8] S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In *Proceedings of the 26th International Conference on Machine Learning*, pages 201–208, Montreal, June 2009.

[9] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556, 2004.

[10] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proc. of the 24th International Conference on Machine Learning*, pages 521–528, 2007.

[11] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, August 2004.

[12] J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society, B*, 66:815–849, 2004.

[13] K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *Annals of Statistics*, 37:1871–1905, 2009.

[14] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with Hilbert-Schmidt norms. *16th International Conf. Algorithmic Learning Theory*, pages 63–77, 2005.

[15] B. Li, H. Zha, and F. Chiaramonte. Contour regression: A general approach to dimension reduction. *The Annals of Statistics*, 33:1580–1616, 2005.

[16] K.-C. Li. Sliced inverse regresion for dimension reduction. *Journal of the American Statistical Association*, 86:316–327, 1991.

[17] U. V. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 5:395–416, 2007.

[18] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX, IEEE*, pages 41–48, 1999.

[19] P. Murphy and D. Aha. UCI repository of machine learning databases. Technical report, University of California, Irvine, 1994.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.

[21] J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimension reduction. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 697–704, 2007.

[22] D. Niu, J. Dy, and M. Jordan. Multiple non-redundant spectral clustering views. In *Proceedings of the 27th International Conference on Machine Learning*, pages 831–838, 2010.

[23] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[24] B. Scholköpf, A. J. Smola, and K. R. Müller. Kernel principal component analysis. In *7th International Conference on Artificial Neural Networks*, pages 583–588, 1997.

[25] L. Song, A. J. Smola, A. Gretton, and K. M. Borgwardt. A dependence maximization view of clustering. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 815–822, 2007.

[26] A. Strehl and J. Ghosh. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.

[27] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[28] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side information. In *Advances in Neural Information Processing Systems, 15*, pages 505–512, 2003.

[29] K. Yeung and W. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.

[30] Z. Zhang and M. I. Jordan. Multiway spectral clustering: A margin-based perspective. *Statistical Science*, 23:383–403, 2008.