Multiple Non-Redundant Spectral Clustering Views

Donglin Niu DNIU@ECE.NEU.EDU
ECE Department, Northeastern University, Boston, MA 02115

Jennifer G. Dy JDY@ECE.NEU.EDU
ECE Department, Northeastern University, Boston, MA 02115

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
EECS and Statistics Departments, University of California, Berkeley, CA 94720

Abstract

Many clustering algorithms only find one clustering solution. However, data can often be grouped and interpreted in many different ways. This is particularly true in the high-dimensional setting where different subspaces reveal different possible groupings of the data. Instead of committing to one clustering solution, here we introduce a novel method that can provide several non-redundant clustering solutions to the user. Our approach simultaneously learns non-redundant subspaces that provide multiple views and finds a clustering solution in each view. We achieve this by augmenting a spectral clustering objective function to incorporate dimensionality reduction and multiple views and to penalize for redundancy between the views.

1. Introduction

Clustering is often a first step in the analysis of complex multivariate data, particularly when a data analyst wishes to engage in a preliminary exploration of the data. Most clustering algorithms find one partitioning of the data (Jain et al., 1999), but this is overly rigid. In the exploratory data analysis setting, there may be several views of the data that are of potential interest. For example, given patient information data, what is interesting to physicians will be different from what insurance companies find interesting. This multi-faceted nature of data is particularly prominent in the high-dimensional setting, where data such as text, images and genotypes may be grouped together in several different ways for different purposes. For example, images of faces of people can be grouped based on their pose or identity. Web pages collected from universities can be clustered based on the type of webpage's owner, {*faculty, student, staff*}, field, {*physics, math, engineering, computer science*}, or identity of the university. In some cases, a data analyst wishes to find a single clustering, but this may require an algorithm to consider multiple clusterings and discard those that are not of interest. In other cases, one may wish to summarize and organize the data according to multiple possible clustering views. In either case, it is important to find multiple clustering solutions which are non-redundant.

Although the literature on clustering is enormous, there has been relatively little attention paid to the problem of finding multiple non-redundant clusterings. Given a single clustering solution, Bae & Bailey (2006) impose cannot-link constraints on data points belonging to the same group in that clustering and then use agglomerative clustering in order to find an alternative clustering. Gondek & Hofmann (2004) use a conditional information bottleneck approach to find an alternative clustering to a particular clustering. Qi & Davidson (2009) propose an approach based on Gaussian mixture models in which they minimize the KL-divergence between the projection of the original empirical distribution of the data and the projection subject to the constraint that the sum-of-squared error between samples in the projected space and the means of the clusters they do not belong to is smaller than a pre-specified threshold. All of these methods find a single alternative view given one clustering solution or a known grouping. In contrast, the approach that we present here can discover multiple (i.e., more than two) views.

Recently, Caruana et al. (2006), Cui et al. (2007) and Jain et al. (2008) also recognized the need to find

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

multiple clustering solutions from data. The meta-clustering method in Caruana et al. (2006) generates a diverse set of clustering solutions by either random initialization or random feature weighting. Then to avoid presenting the user with too many clusterings, these solutions are combined using agglomerative clustering based on a Rand index for measuring similarity between pairwise clustering solutions. Our approach differs from meta-clustering in that we directly seek out multiple solutions by optimizing a multiple non-redundant clustering criterion rather than relying on random initialization or random feature weighting. Cui et al. (2007) propose a sequential method that starts by finding a dominant clustering partition, and then finds alternative views by clustering in the subspace orthogonal to the clustering solutions found in previous iterations. Jain et al. (2008) propose a non-sequential method that learns two disparate clusterings simultaneously by minimizing a K-means sum-squared error objective for the two clustering solutions while at the same time minimizing the correlation between these two clusterings. Both of these methods are based on K-means and are thus limited to convex clusters. In contrast, the approach we introduce here can discover non-convex shaped clusters in each view; we view this capability as important in the exploratory data analysis setting. Moreover, the method in Jain et al. (2008) uses all the features in all views. Our approach is based on the intuition that different views most likely exist in different subspaces and thus we learn multiple subspaces in conjunction with learning the multiple alternative clustering solutions.

In summary, this work that we present here advances the field in the following way: (1) we study an important multiple clustering discovery paradigm; (2) within this paradigm, we develop a novel approach that can find clusters with arbitrary shapes in each view; (3) within each view, our method can learn the subspace in which the clusters reside; and finally, (4) we simultaneously learn the multiple subspaces and the clusterings in each view by optimizing a single objective function.

2. Formulation

Our goal is to find multiple clustering views. Given n data samples, there are c^n possible c disjoint partitionings of the data (counting permutations of the same groupings). Only a small number of these groupings are likely to be meaningful. We would like the clusters in each view to be of *good quality* and we also wish for the clustering solutions in the different views to provide *non-redundant* information so as not to overwhelm the data analyst. Moreover, typically different views or ways of grouping reside in different subspaces; thus, we wish to incorporate *learning of the subspace* in which the clusters lie in each view as well.

To obtain high-quality clusterings, we base our approach on a spectral clustering formulation (Ng et al., 2001); the spectral approach has the advantage that it avoids strong assumptions on cluster shapes. This creates a challenge for the design of the measure of dependence among views in that we must be able to measure non-linear dependencies. We make use of the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005) for this purpose. That is, we use the HSIC as a penalty that is added to our spectral clustering criterion. HSIC measures the statistical dependence among views and drives the learning algorithm toward finding views that are as independent from each other as possible. We now provide a fuller description of the main ingredients of our algorithm.

1. Cluster Quality and Spectral Clustering. There are many ways to define the quality of clusters resulting in a variety of clustering algorithms in the literature (Jain et al., 1999). In this paper, we focus on spectral clustering because it is a flexible clustering algorithm that is applicable to different types of data and makes relatively weak assumptions on cluster shapes (clusters need not be convex or homogeneous). There are several ways to explain spectral clustering (Von Luxburg, 2007). Here, we present the graph partitioning viewpoint. Given a set of n data samples, $\{x_1, \ldots, x_n\}$, with each x_i be a column vector in $\mathbb{R}^d$, let $k(\cdot, \cdot) \geq 0$ be a kernel function that measures some notion of similarity between data points. We let $k_{ij} = k(x_i, x_j)$ denote the kernel function evaluated at points x_i and x_j. To obtain flexible cluster shapes, we use nonlinear kernel functions such as polynomial and Gaussian kernels. Let $G = \{V, E\}$ be a graph, with $V = \{v_1, \ldots, v_n\}$ as the set of vertices and E as the set of edges connecting the vertices. Each vertex v_i in this graph represents a data point x_i. The edge weights between pairs of vertices (v_i and v_j) are defined by k_{ij}. Let K be the similarity matrix with elements k_{ij}. The goal of clustering is to partition data $\{x_1, \ldots, x_n\}$ into c disjoint partitions, $P_1, \ldots, P_c$. We would like the similarity of the samples *between groups* to be low, and similarity of the samples *within groups* to be high. There are several varieties of graph partitioning objective functions. In this paper, we make use of the c-way normalized cut objective, $NCut(G)$, defined as follows: $NCut(P_1, ..., P_c) = \sum_{t=1}^{c} \frac{cut(P_t, V \backslash P_t)}{vol(P_t)}$ 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 $V \backslash \mathcal{A}$ is the complement of $\mathcal{A}$. Optimizing this objective function is an NP-hard discrete optimization problem, thus spectral clustering relaxes the discreteness of the indicator matrix and allows its entries to take on any real value. If we let U denote this relaxed indicator matrix, of size n by c, the relaxed optimization problem reduces to the following trace maximization problem:

$$\begin{array}{rl} \max_{U \in R^{n \times c}} & \mathrm{tr}(U^T D^{-1/2} K D^{-1/2} U) \\ \text{s.t.} & U^T U = I. \end{array} \qquad (1)$$

where $\mathrm{tr}(\cdot)$ is the trace function, D is a diagonal matrix with diagonal elements equal to d_i, and I is the identity matrix. The solution U to this optimization problem involves taking the first c eigenvectors corresponding to the largest c eigenvalues of the normalized similarity matrix $L = D^{-1/2} K D^{-1/2}$. To obtain the discrete partitioning of the data, we re-normalize each row of U to have unit length and then apply K-means to each row of the normalized U. We assign each x_i to the same cluster that the row u_i is assigned to. This particular version of spectral clustering is due to Ng et al. (2001).

2. Learning the Low-Dimensional Subspace. Our goal is to find m low-dimensional subspaces, where m is the number of views, such that in each view, clusters are well-separated (linearly or nonlinearly). We learn the subspace in each view by coupling dimensionality reduction with spectral clustering in a single optimization objective. In each view, instead of utilizing all the features/dimensions in computing the kernel similarity matrix K, similarity is computed in subspace W_q: our algorithm is based on the kernel function $k(W_q^T x_i, W_q^T x_j)$, where $W_q \in R^{d \times l_q}$ is a transformation matrix for each view that transforms $x_i \in R^d$ in the original space to a lower-dimensional space l_q ($l_q \leq d$, $\sum_q l_q \leq d$).

3. How to Measure Redundancy. One way to measure redundancy between two variables is in terms of their correlation coefficient; however, this captures only linear dependencies among random variables. Another approach involves measuring the mutual information, but this requires estimating the joint distribution of the random variables. Recent work by Fukumizu et al. (2009) and Gretton et al. (2005) provide a way to measure dependence among random variables without explicitly estimating joint distributions and without having to discretize continuous random variables. 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. Consider $\mathcal{X}$ and $\mathcal{Y}$ to be two sample spaces with random variables (x, y) drawn from these spaces. Let us define a mapping $\phi(x)$ from $x \in \mathcal{X}$ to kernel space $\mathcal{F}$, such that the inner product between vectors in that space is given by a kernel function, $k_1(x, x') = \langle \phi(x), \phi(x') \rangle$. Let $\mathcal{G}$ be a second kernel space on $\mathcal{Y}$ with kernel function $k_2(\cdot, \cdot)$ and mapping $\varphi(y)$. A linear cross-covariance operator $C_{xy} : \mathcal{G} \to \mathcal{F}$ between these feature maps is defined as: $C_{xy} = E_{xy}[(\phi(x) - \mu_x) \otimes (\varphi(y) - \mu_y)]$, where $\otimes$ is the tensor product. Based on this operator, Gretton et al. (2005) define the Hilbert-Schmidt independence criterion (HSIC) between two random variables, x and y, as follows:

$$\begin{array}{rl} \mathrm{HSIC}(p_{xy}, \mathcal{F}, \mathcal{G}) = & \|C_{xy}\|^2_{\mathrm{HS}} \\ = & E_{x,x',y,y'}[k_1(x,x') k_2(y,y')] + \\ & E_{x,x'}[k_1(x,x')] E_{y,y'}[k_2(y,y')] \\ & -2 E_{x,y}[E_{x'}[k_1(x,x')] E_{y'}[k_2(y,y')]]. \end{array}$$

Given n observations, $Z := \{(x_1, y_1), ..., (x_n, y_n)\}$, we can empirically estimate the HSIC by:

$$\mathrm{HSIC}(Z, \mathcal{F}, \mathcal{G}) = (n-1)^{-2} \mathrm{tr}(K_1 H K_2 H) \qquad (2)$$

where $K_1, \ K_2 \in R^{n \times n}$ are Gram matrices, $(K_1)_{ij} = k_1(x_i, x_j)$, $(K_2)_{ij} = k_2(y_i, y_j)$, and where $(H)_{ij} = \delta_{ij} - n^{-1}$ centers the Gram matrices to have zero mean in the feature space. We use the HSIC as a penalty term in our objective function to ensure that subspaces in different views provide non-redundant information.

2.1. Overall Multiple Non-Redundant Spectral Clustering Objective Function

For each view q, $q = 1, \ldots, m$, let W_q be the subspace transformation operator, U_q be the relaxed cluster membership indicator matrix, K_q be the Gram matrix, and D_q be the corresponding degree matrix for that view. Our overall objective function, f, is:

$$\begin{array}{rl} \max_{U_1 \ldots U_m, W_1 \ldots W_m} & \sum_q \mathrm{tr}(U_q^T D_q^{-1/2} K_q D_q^{-1/2} U_q) \\ & -\lambda \sum_{q \neq r} \mathrm{HSIC}(W_q^T x, W_r^T x) \\ \text{s.t.} & U_q^T U_q = I \\ & (K_q)_{ij} = k_q(W_q^T x_i, W_q^T x_j) \\ & W_q^T W_q = I. \end{array} \qquad (3)$$

The first term $\sum_q \mathrm{tr}(U_q^T D_q^{-1/2} K_q D_q^{-1/2} U_q)$ is the relaxed spectral clustering objective in Eq. (1) for each view and it optimizes for cluster quality. In the second term, $\sum_{q \neq r} \mathrm{HSIC}(W_q^T x, W_r^T x)$ from Eq. (2) is used to penalize for dependence among subspaces in different views. Simply optimizing one of these criteria is not enough to produce quality non-redundant multiple clustering solutions. Optimizing the spectral criterion alone can still end up with redundant clusterings.

Optimizing HSIC alone leads to an independent subspace analysis problem (Theis, 2007), which can find views with independent subspaces but data in these subspaces may not lead to good clustering solutions. The parameter λ is a regularization parameter that controls the trade-off between these two criteria. As a rule of thumb, we suggest choosing a value of λ that makes the first and second term to be of the same order.

2.2. Algorithm

In this section, we describe how we optimize our overall objective function formulation in Eq. (3). The optimization is carried out in two steps:

Step 1: Assuming all W_q fixed, we optimize for U_q in each view.
With the projection operators W_q fixed, we can optimize the similarity and degree matrices K_q and D_q for each view respectively. Similar to spectral clustering, here we relax the indicator matrix U_q to range over real values. The problem now becomes a continuous optimization problem resulting in an eigenvalue problem. The solution for U_q is equal to the first c_q eigenvectors (corresponding to the largest c_q eigenvalues) of the matrix $D_q^{-1/2} K_q D_q^{-1/2}$, where c_q is the number of clusters for view q. Then we normalize each row of U_q to have unit length. Note that unlike applying spectral clustering on the projected space $W_q^T x$, this optimization step stops here; it keeps U_q real-valued and does not need to explicitly assign the cluster membership to the samples.

Step 2: Assuming all U_q fixed, we optimize for W_q for each view.
We optimize for W_q by applying gradient ascent on the Stiefel manifold (Edelman et al., 1999; Bach & Jordan, 2002) to satisfy the orthonormality constraints, $W_q^T W_q = I$, in each step. We project the gradient of the objective function onto the tangent space, $\Delta W_{Stiefel} = \frac{\partial f}{\partial W_q} - W_q (\frac{\partial f}{\partial W_q})^T W_q$, which shows that $W_q^T \Delta W_{Stiefel}$ is skew symmetric. We thus update W_q on the geodesic in the direction of the tangent space as follows:

$$W_{new} = W_{old} \exp(\tau W_{old}^T \Delta W_{Stiefel}), \tag{4}$$

where exp means matrix exponential and τ is the step size. We apply a backtracking line search to find the step size according to the Armijo rule to assure improvement of our objective function at every iteration.

The derivative $\frac{\partial f}{\partial W_q}$ is calculated as follows. $L_q = D_q^{-1/2} K_q D_q^{-1/2}$ is the normalized similarity matrix for each view. Letting $k_{q,ij}$ denote the (i,j)th entry in K_q, and letting $d_{q,ii}$ denote the (i,i)th diagonal element in D_q, each element in matrix L_q is $l_{q,ij} = d_{q,ii}^{-1/2} k_{q,ij} d_{q,jj}^{-1/2}$. For a fixed data embedding, the spectral objective can be expressed as a linear combination of each element in matrix L_q with coefficient $u_{q,i} u_{q,j}^T$, where $u_{q,i}$ is the spectral embedding for x_i in view q. Applying the chain rule, the derivative of the element $l_{q,ij}$ with respect to W_q can be expressed as

$$\begin{aligned} l'_{q,ij} = \; & k'_{q,ij} d_{q,ii}^{-\frac{1}{2}} d_{q,jj}^{-\frac{1}{2}} - \tfrac{1}{2} d_{q,ii}^{-\frac{1}{2}} d'_{q,ii} k_{q,ij} d_{q,jj}^{-\frac{1}{2}} \\ & - \tfrac{1}{2} d_{q,jj}^{-\frac{1}{2}} d'_{q,jj} k_{q,ij} d_{q,ii}^{-\frac{1}{2}}, \end{aligned} \tag{5}$$

where $k'_{q,ij}$, $d'_{q,ii}$ and $d'_{q,jj}$ are derivatives of the similarity and degree with respect to W_q. For each view, the empirical HSIC estimate term is not dependent on the spectral embedding U_q and can be expanded as

$$\text{HSIC}(W_q^T x, W_r^T x) = (n-1)^{-2} \text{tr}(K_q H K_r H). \tag{6}$$

If we expand the trace in the HSIC term,

$$\begin{aligned} \text{tr}(K_q H K_r H) = \; & \text{tr}(K_q K_r) - 2n^{-1} \mathbf{1}^T K_q K_r \mathbf{1} \\ & + n^{-2} \text{tr}(K_q) \text{tr}(K_r), \end{aligned} \tag{7}$$

where $\mathbf{1}$ is the vector of all ones. The partial derivative of the two terms in the objective function with respect to W_q is now expressed as a function of the derivative of the kernel function. For example, if we use a Gaussian kernel defined as $k(W_q^T x_i, W_q^T x_j) = \exp(-\left\| W_q^T \Delta x_{ij} \right\|^2 / 2\sigma^2)$, where Δx_{ij} is $x_i - x_j$, the derivative of $k_{q,ij}$ with respect to W_q is

$$\frac{\partial k_{q,ij}}{\partial W_q} = -\frac{1}{\sigma^2} \Delta x_{ij} \Delta x_{ij}^T W_q \exp \frac{-\Delta x_{ij}^T W_q W_q^T \Delta x_{ij}}{2\sigma^2}. \tag{8}$$

We repeat these two steps iteratively until convergence. We set the convergence threshold to be $\varepsilon = 10^{-4}$ in our experiments. After convergence, we obtain the discrete clustering solutions by using the standard K-means step of spectral clustering in the embedding space U_q in each view. Algorithm 1 provides a summary of our approach.

2.3. Implementation Details

In this section we describe some practical implementation details for our algorithm.

Initialization. Our algorithm can get stuck at a local optimum, making it dependent on initialization. We would like to start from a good initial guess. We initialize the subspace views W_q by clustering the features, such that features assigned to the same views are dependent on each other and those in different views are as independent from each other as possible. We

Algorithm 1 Multiple Spectral Clustering

```
**Input:** Data $x$, cluster number $c_q$ for each view and number of views $m$.
**Initialize:** All $W_q$ by clustering the features.
**Step 1:** For each view $q$, project data on subspaces $W_q$, $q = 1, \ldots, m$.
Calculate the kernel similarity matrix $K_q$ and degree matrix $D_q$ in each subspace.
Calculate the top $c_q$ eigenvectors of $L_q = D_q^{-1/2} K_q D_q^{-1/2}$ to form matrix $U_q$. Normalize rows of $U_q$ to have unit length.
**Step 2:** Given all $U_q$, update $W_q$ based on gradient ascent on the Stiefel manifold.
**REPEAT** steps 1 and 2 until convergence.
**K-means Step:** Form $n$ samples $u_{q,i} \in R^{c_q}$ from rows of $U_q$ for each view. Cluster the points $u_{q,i}$, $i = 1, \ldots, n$, using $K$-means into $c_q$ partitions, $P_1, \ldots, P_{c_q}$.
**Output:** Multiple clustering partitions and transformation matrices $W_q$.
```

measure dependence based on HSIC. First, we calculate the similarity, a_{ij}, of each pair of features, f_i and f_j, using HSIC, to build a similarity matrix A. For discrete features, similarity is measured by normalized mutual information. Second, we apply spectral clustering (Ng et al., 2001) using this similarity matrix to cluster the features into m clusters, where m is the number of desired views. Each feature cluster q corresponds to our view q. We initialize each subspace view W_q to be equivalent to the projection that selects only the features in cluster q. We build W_q as follows. For each feature j in cluster q, we append a column of size d by 1 to W_q whose entries are all zero except for the jth element which is equal to one. This gives us matrix W_q of size d by l_q, where d is the original dimensionality and l_q the number of features assigned to cluster q. We find this is a good initialization scheme because this provides us with multiple subspaces that are approximately as independent from each other as possible. Additionally, this scheme provides us with an automated way of setting the dimensionality for each view l_q. Although we start with disjoint features, the final learned W_q in each view are transformation matrices, where each feature can have some weight in one view and some other weight in another view.

Kernel Similarity Approximation. Calculating the kernel similarity matrix K is time consuming. We apply incomplete Cholesky decomposition as suggested in Bach & Jordan (2002), giving us an approximate kernel similarity matrix $\widetilde{K}$. Using incomplete Cholesky decomposition, the complexity of calculating the kernel matrix is $O(ns^2)$, where n is the number of data instances, s is the size of the approximation matrix $\widetilde{G}$, where $\widetilde{K} = \widetilde{G}\widetilde{G}^T$. Thus, the complexities of our derivative computation and eigen-decomposition are now $O(nsd)$ and $O(ns^2)$ respectively.

3. Experiments

We performed experiments on both synthetic and real data to investigate the capability of our algorithm to yield reasonable non-redundant multiple clustering solutions. In particular, we present the results of experiments on two synthetic data and four real data: a corpus of face image data, a corpus of machine sounds and two text data sets. We compare our method, *multiple SC* (mSC), to two recently proposed algorithms for finding multiple clusterings: *orthogonal projection clustering* (OPC) (Cui et al., 2007) and *de-correlated K-means* (DK) (Jain et al., 2008). We also compare against standard spectral clustering (SC) and standard K-means. In these standard algorithms, different views are generated by setting K to the number of clusters in that view. In orthogonal projection clustering (Cui et al., 2007), instances are clustered in the principal component space (retaining 90% of the total variance) by a suitable clustering algorithm to find a dominant clustering. Then data are projected to the subspace that is orthogonal to the subspace spanned by the means of the previous clusters. This process is repeated until all the possible views are found. In de-correlated K-means (Jain et al., 2008), the algorithm simultaneously minimizes the sum-of-squared errors (SSEs) in two clustering views and the correlation of the mean vectors and representative vectors between the two views. Gradient descent is then used to find the clustering solutions. In this approach, both views minimize SSEs in all the original dimensions. We set the number of views and clusters in each view equal to the known values for all methods.

We measure the performance of our clustering methods based on the normalized mutual information (NMI) (Strehl & Ghosh, 2002) between the clusters found by these methods with the "true" class labels. Let A represent the clustering results and B the labels, $NMI = \frac{H(A)-H(A|B)}{\sqrt{H(A)H(B)}}$, where $H(\cdot)$ is the entropy. Note that in all our experiments, labeled information is not used for training. We only use the labels to measure the performance of our clustering algorithms. Higher NMI values mean that the clustering results are more similar to the labels; the criterion reaches its maximum value of one when the clustering and labels are perfectly matched. To account for randomness in the algorithms, we report the average NMI results and their standard deviations over ten runs. For multiple clustering methods, we find the best matching partitioning and view based on NMI and report that NMI. In all of our experiments we use a Gaussian kernel, except for the text data where we use a polynomial kernel. We set the kernel parameters so as to obtain the maximal eigen-gap between the kth and $k+1$th eigen-value for the matrix L. The regularization parameter λ was set in the range $0.5 < \|\frac{\lambda \mathrm{HSIC}}{\mathrm{tr}(U^T LU)}\| < 1.5$.

3.1. Results on Synthetic Data

Table 1. NMI Results for Synthetic Data

	Data 1		Data 2	
	view 1	view 2	view 1	view 2
mSC	**.94±.01**	**.95±.02**	**.90±.01**	**.93±.02**
OPC	.89±.02	.85±.03	.02±.01	.07±.03
DK	.87±.03	.94±.03	.03±.02	.05±.03
SC	.37±.03	.42±.04	.31±.04	.25±.04
K-means	.36±.03	.34±.04	.03±.01	.05±.02

Our first experiment was based on a synthetic data set consisting of two alternative views to which noise features were added. There were six features in total. Three Gaussian clusters were generated in the feature subspace $\{f_1, f_2\}$ as shown in Figure 1(a). The color and symbols of the points in Figure 1 indicate the cluster labeling in the first view. The other three Gaussian clusters were generated in the feature subspace $\{f_3, f_4\}$ displayed in Figure 1(b). The remaining two features were generated from two independent Gaussian noise with zero mean and variance $\sigma^2 = 25$. Here, we test whether our algorithm can find the two views even in the presence of noise. The second synthetic data set has two views with arbitrarily shaped clusters from four dimensions. The two clustering views are in the two subspaces $\{f_1, f_2\}$ and $\{f_3, f_4\}$, respectively, as shown in Figure 1(c) and Figure 1(d). In this data set, we investigate whether or not our approach can discover arbitrarily shaped clusters in alternative clustering views. Table 1 presents the average NMI values obtained by the different methods for the different views on these synthetic data. The best values are highlighted in bold font. The results in Table 1 show that our approach works well on both data sets. Orthogonal clustering and de-correlated K-means both performed poorly on synthetic data set 2 because they are not capable of discovering clusters that are non-spherical. Note that standard spectral clustering and K-means also performed poorly because they are designed to only search for one clustering solution. Standard SC was better than all of the K-means based method for synthetic data set 2, but it is still far worse than our proposed mSC algorithm, which can discover multiple arbitrarily shaped clusterings simultaneously.

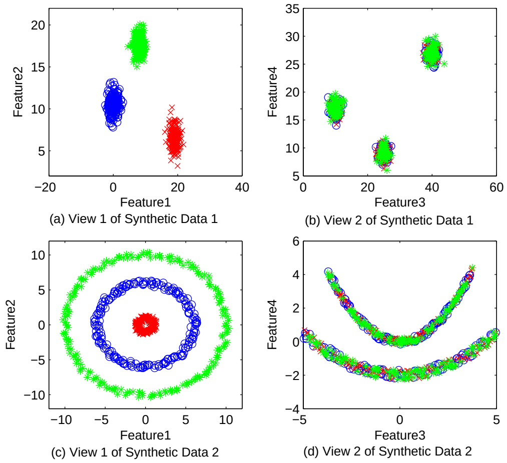


Figure 1. (a) View 1 and (b) View 2 of synthetic data set 1; (c) View 1 and (d) View 2 of synthetic data set 2.

3.2. Results on Real Data

We now test our method on four real-world data sets to see whether we can find meaningful clustering views. We selected data that are high dimensional and intuitively are likely to present multiple possible partitionings. In particular, we test our method on face image, a sound data set and two text data sets. Table 2 presents the average NMI results for the different methods on the different clustering/labeling views for these real data sets.

Face Data. The face data set from the UCI KDD archive (Bay, 1999) 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). The image resolution is 32 × 30, resulting in a data set with 640 instances and 960 features. The two dominant views inferred from this data are identity and pose. Figure 2 shows the mean face image for each cluster in two clustering views. The number below each image is the percentage of this person appearing in this cluster. Note that the first view captures the identity of each person, and the second view captures the pose of the face images. Table 2 reveals that our approach performed the best (as shown in bold) in terms of NMI compared to the other two competing methods and also compared to standard SC and K-means.

Table 2. NMI Results for Real Data

	Face		Machine Sound			WebKB Data	
	ID	Pose	Motor	Fan	Pump	Univ	Owner
mSC	**0.79±0.03**	**0.42±0.03**	**0.82±0.03**	**0.75±0.04**	**0.83±0.03**	**0.81±0.02**	0.54±0.04
OPC	0.67 ± 0.02	0.37 ± 0.01	0.73 ± 0.02	0.68 ± 0.03	0.47 ± 0.04	0.43 ± 0.03	0.53 ± 0.02
DK	0.70 ± 0.03	0.40 ± 0.04	0.64 ± 0.02	0.58 ± 0.03	0.75 ± 0.03	0.48 ± 0.02	**0.57±0.04**
SC	0.67 ± 0.02	0.22 ± 0.02	0.42 ± 0.02	0.16 ± 0.02	0.09 ± 0.02	0.25 ± 0.02	0.39 ± 0.03
K-means	0.64 ± 0.04	0.24 ± 0.04	0.57 ± 0.03	0.16 ± 0.02	0.09 ± 0.02	0.10 ± 0.03	0.50 ± 0.04

Machine Sound Data. In this section, we report results of an experiment on the classification of acoustic signals inside buildings into different machine types. We collected sound signals with accelerometers, yielding a library of 280 sound instances. Our goal is to classify these sounds into three basic machine classes: `motor, fan, pump`. Each sound instance can be from one machine, or from a mixture of two or three machines. As such, this data has a multiple clustering view structure. In one view, data can be grouped as `motor` or `no motor`; the other two views are similarly defined. 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. Table 2 shows that our method outperforms orthogonal projection clustering, de-correlated K-means, standard SC, and standard K-means. We performed much better than the competing methods probably because we can find independent subspaces and arbitrarily shaped clusters simultaneously.

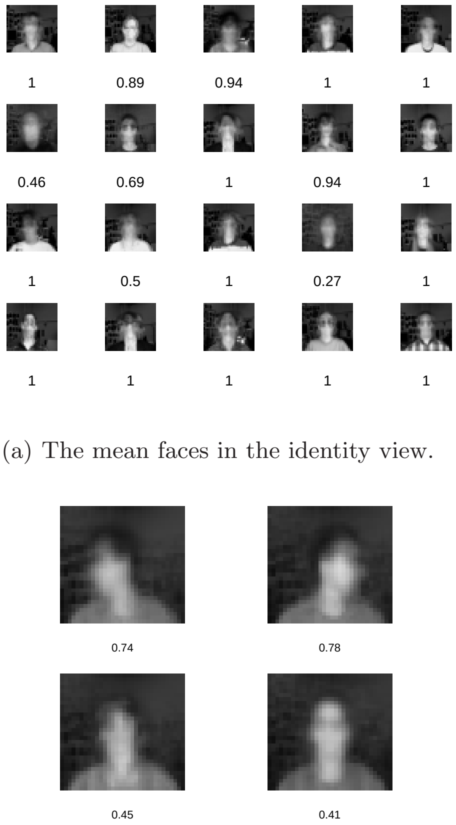
WebKB Text Data. This data set[1] contains html documents from four universities: Cornell University, University of Texas, Austin, University of Washington and University of Wisconsin, Madison. We removed the miscellaneous pages and subsampled a total of 1041 pages from four web-page owner types: course, faculty, project and student. We preprocessed the data by removing rare words, stop words, and words with small variances, giving us a total of 350 words in the vocabulary. Average NMI results are shown in Table 2. mSC is the best in discovering view 1 based on universities (with NMI values around 0.81, while the rest are ≤ 0.48), and comes in close second to decorrelated-K-means in discovering view 2 based on owner types (0.54 and 0.57 respectively). A possible reason why we do much better than the other approaches in view 1 is because we can capture nonlinear dependencies among views, whereas OPC and DK only consider linear dependencies. In this data set, the two clustering views (universities and owner) reside in two different feature subspaces. Our algorithm, mSC, also discovered these subspaces correctly. In the university view, the five highest variance features we learned are: {*Cornell, Texas, Wisconsin, Madison, Washington*}. In the type of web-page owner view, the highest variance features we learned are: {*homework, student, professor, project, ph.d*}.

NSF Text Data. The NSF data set (Bay, 1999) consists of $129,000$ abstracts from year 1990 to 2003. Each text instance is represented by the frequency of occurrence of each word. We select 1000 words with the highest frequency variance in the data set and randomly subsample 15000 instances for this experiment. Since this data set has no labels, we do not report any NMI scores; instead, we use the five highest frequency words in each cluster to assess what we discovered. We observe that view 1 captures the type of research: theoretical research in cluster 1 represented by words: {*methods, mathematical, develop, equation, problem*} and experimental research in cluster 2 represented by words: {*experiments, processes, techniques, measurements, surface*}. We observe that view 2 captures different fields: materials, chemistry and physics in cluster 1 by words: {*materials, chemical, metal, optical, quantum*}, control, information theory and computer science in cluster 2 by words: {*control, programming, information, function, languages*}, and biology in cluster 3 by words: {*cell, gene, protein, dna, biological*}.

4. Conclusions

We have introduced a new method for discovering multiple non-redundant clustering views for exploratory data analysis. Many clustering algorithms only find a single clustering solution. However, data may be multi-faceted by nature; also, different data analysts may approach a particular data set with different goals in mind. Often these different clusterings reside in different lower dimensional subspaces. To address these issues, we have introduced an optimization-based framework which optimizes both a spectral clustering objective (to obtain high-quality clusters) in each sub-

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/



(a) The mean faces in the identity view.

(b) The mean faces in the pose view.

Figure 2. Multiple non-redundant spectral clustering results for the face data set.

space, and the HSIC objective (to minimize the dependence of the different subspaces). The resulting mSC method is able to discover multiple non-redundant clusters with flexible cluster shapes, while simultaneously finding low-dimensional subspaces in each view. Our experiments on both synthetic and real data show that our algorithm outperforms competing multiple clustering algorithms (orthogonal projection clustering and de-correlated K-means).

Acknowledgments

This work is supported by NSF IIS-0915910.

References

Bach, F. R. and Jordan, M. I. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Bae, E. and Bailey, J. COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *IEEE International Conference on Data Mining*, pp. 53–62, 2006.

Bay, S. D. The UCI KDD archive, 1999. URL http://kdd.ics.uci.edu.

Caruana, R., Elhawary, M., Nguyen, N., and Smith, C. Meta clustering. In *IEEE International Conference on Data Mining*, pp. 107–118, 2006.

Cui, Y., Fern, X. Z., and Dy, J. Non-redundant multi-view clustering via orthogonalization. In *IEEE Intl. Conf. on Data Mining*, pp. 133–142, 2007.

Edelman, A., Arias, T. A., and Smith, S. T. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

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

Gondek, D. and Hofmann, T. Non-redundant data clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 75–82, 2004.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with hilbert-schmidt norms. *16th International Conf. Algorithmic Learning Theory*, pp. 63–77, 2005.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Computing Surveys*, 31 (3):264–323, 1999.

Jain, P., Meka, R., and Dhillon, I. S. Simultaneous unsupervised learing of disparate clustering. In *SIAM Intl. Conf. on Data Mining*, pp. 858–869, 2008.

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

Qi, Z. J. and Davidson, I. A principled and flexible framework for finding alternative clusterings. In *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2009.

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

Theis, F. J. Towards a general independent subspace analysis. In *Advances in Neural Information Proc. Systems*, volume 19, pp. 1361–1368, 2007.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 5:395–416, 2007.