DSP: Robust Semi-Supervised Dimensionality Reduction using Dual Subspace Projections

Su Yan*
IBM Almaden Research Center
San Jose, CA, 95120, USA
syan@us.ibm.com

Sofien Bouaziz
Ecole Polytechnique Fédérale de Lausanne
Switzerland
sofien.bouaziz@gmail.com

Dongwon Lee
Pennsylvania State University
University Park, PA 16802, USA
dongwon@psu.edu

Abstract—**High-dimensional data usually incur learning deficiencies and computational difficulties. We present a novel semi-supervised dimensionality reduction technique that embeds high-dimensional data in an optimal low-dimensional subspace, which is learned with a few user supplied constraints as well as the structure of input data. We study two types of constraints that indicate whether or not pairs of data points originate from the same class. Data partitions that satisfy both types of constraints may be conflicting. To solve this problem, our method projects data into two different subspaces, one in the kernel space and one in the original input space, each is designed for enforcing one type of constraints. Projections in the two spaces interact and data are embedded in an optimal low-dimensional subspace where constraints are maximally satisfied. Besides constraints, our method also preserves the intrinsic data structure, such that nearby/far away data points in the original space are still near to/far from each other in the embedded space. Compared to existing techniques, our method has the following advantages: 1) It can benefit from constraints even when only a few are available. 2) It is robust and does not suffer from overfitting. 3) It handles nonlinearly separable data, but learns a linear data transformation. Thus the method can be easily generalized to new data points and is efficient in dealing with large data sets. Experiments on real data from multiple domains clearly demonstrate that significant improvements in learning accuracy can be achieved after dimensionality reduction by employing only a few constraints.**

I. Introduction

There are two major difficulties in analyzing or learning from high-dimensional data. First, the learning accuracy is low due to the redundancies in high-dimensional feature spaces and the relatively small amount of training available compared to the dimensionality. Second, the computational cost is so high that many techniques are not readily applicable to handle large amount of high-dimensional data [9].

Dimensionality reduction is the technique that solves the high dimensionality problem and has been extensively studied and widely applied in text categorization, face recognition and microarray gene expression analysis where data are usually expressed as vectors of high dimensionality. Two representative dimensionality reduction techniques are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA is an unsupervised method that maximally preserves the variance of the data, and LDA is a supervised method that achieves maximal class separation by maximizing the ratio of between-class variance to the within-class variance. Both PCA and LDA are *global* methods since they do not preserve local data structures. To overcome the drawbacks of global methods and their variants, a number of *local* dimensionality reduction methods have been proposed, such as Locally Linear Embedding (LLE) [7] and Locality Preserving Projections (LPP) [5]. These methods embed data in the low-dimensional space such that nearby data points in the original space are still near to each other in the embedded space.

Recently, *semi-supervised dimensionality reduction* has stirred many research interests. This is due to the fact that supervision in the form of pairwise constraints is often easier to get than labeled data, and is naturally available in many real application domains. For example, it may be difficult, tedious or costly for users to label thousands of images into pre-set class labels. However, when users are presented with a few simple binary questions of the form "are objects in image a and b the same?", answering Yes/No to the questions is a lot easier. Moreover, for the task of Web document clustering, documents which share large number of similar hyperlinks, or a group of documents with strong co-citation (i.e., co-reference) patterns can be viewed as similar in content.

*Su Yan was at Pennsylvania State University during this work.

Constraints take two general forms: the *must-links* are pairs of points that originate from the same class and thus should be grouped together, and the *cannot-links* are pairs of points that should be put into different groups. To incorporate constraints in dimensionality reduction, [2] proposed relevant component analysis (RCA) that exploits must-links only. [6] extends RCA by exploring cannot-links. Recently, [1] proposed to incorporate constraints using a modified LPP cost function. All these methods exploit constraints only and do not consider the usefulness of abundant unconstrained data. With limited constraints, the methods face the overfitting problem. That is, the subspace that best satisfies a few pairs of constraints does not necessarily reveal the structure of the entire dataset. To this end, [10] and [3] proposed semi-supervised dimensionality reduction methods that exploit both constraints and unconstrained data. However, both methods need users to intuitively set parameters to balance the constrained and the unconstrained data. Besides, all the aforementioned existing methods for semi-supervised dimensionality reduction have their kernel-space equivalents to deal with non-linearly separable data. However, because the projection is non-linear, in order to compute the projection of testing points all the training points besides the transformation matrix need to be stored, and the inner product between the testing points to all the training points need to be calculated and stored. Therefore, such extra storage and computational cost limit their application to large datasets.

In this paper, we propose a novel semi-supervised dimensionality reduction technique named as DSP (Dual Subspace Projections) which can simultaneously preserve the structure of original high-dimensional data and the pairwise constraints specified by users. Thus, the method does not overfit. Furthermore, our method has a closed-form solution of an generalized eigenvalue problem, and therefore can be solved efficiently in the training phase. Moreover, the method uses kernel trick to handle nonlinearly separable data, but the learned projection is still linear. So handling testing data is very efficient.

II. Method Overview

The motivation in this paper is to enforce a set of pairwise constraints in dimensionality reduction such that the intrinsic structure of data in the reduced space can be easily captured by the following data analysis phases, such as clustering and classification. Without loss of generality, we evaluate our dimensionality-reduction technique for clustering tasks, although the technique is equally applicable to classification problems too.

The two types of constraints often lead to conflicting data partitions, even if constraints by themselves are consistent. This is because data are not linearly separable in the input space. The problem can be solved by using the kernel trick. It is always possible to find a data partition that satisfies all the constraints in the high-dimensional space. However, kernel machine will overfit with limited constraints.

Our proposed method alleviates the conflicting constraints problem by exploiting two types of constraints separately in two different subspaces. First, data points are projected to a high-dimensional kernel space, where we further embed data to a subspace such that the two data points constrained by a must-link will be mapped to a single point. This idea originates from [8], where must-link constraints are explored to improve kernel Mean Shift clustering performance. Second, the pairwise distances of embedded data are further explored in the original input space. In particular, we enforce the cannot-link constraints and the intrinsic structure of the input data at this step. We embed data into the second subspace such that nearby/far away data points in the original input space are still near to/far from each other. Besides, cannot-linked data points are also projected to be well separated. The second subspace is therefore a desirable projection direction since it embodies both types of constraints as well as the original data structure.

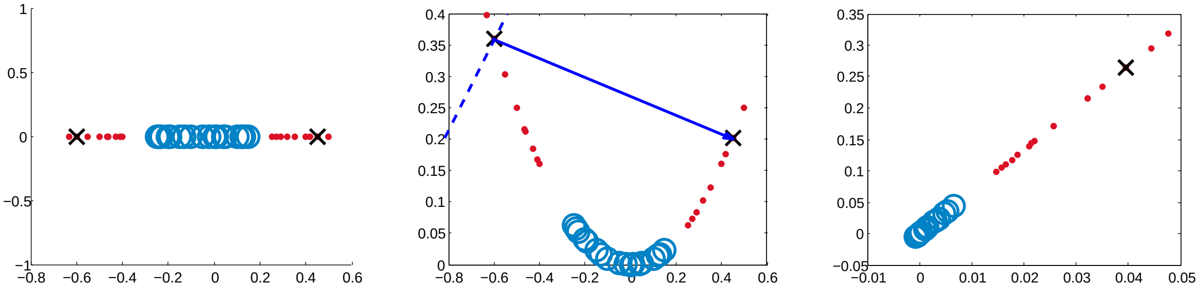
III. Main Proposal

A. Problem Setting

Let $\mathcal{X}$ be the input space containing n data points in f dimensions, $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$. We are given two types of pairwise constraints organized in two sets. Let $\Omega_M = \{(\mathbf{x}_i, \mathbf{x'}_i)\}_{i=1}^{m}$ be the set of m pairs of must-link constraints, and $\Omega_C = \{(\mathbf{x}_i, \mathbf{x'}_i)\}_{i=1}^{c}$ be the set of c pairs of cannot-link constraints. Let r be a desired subspace dimensionality. We want to embed the f-dimensional data in an r-dimensional subspace, s.t. $r \ll f$ by learning a linear data transformation $\mathbf{Z} \in \mathbb{R}^{f \times r}$, such that $\mathbf{y} = \mathbf{Z}^T\mathbf{x}$ where $\mathbf{y}$ is the low-dimensional embedding of $\mathbf{x}$. The Euclidean distance between two points $\mathbf{y}_1$ and $\mathbf{y}_2$ in the reduced space can be expressed as

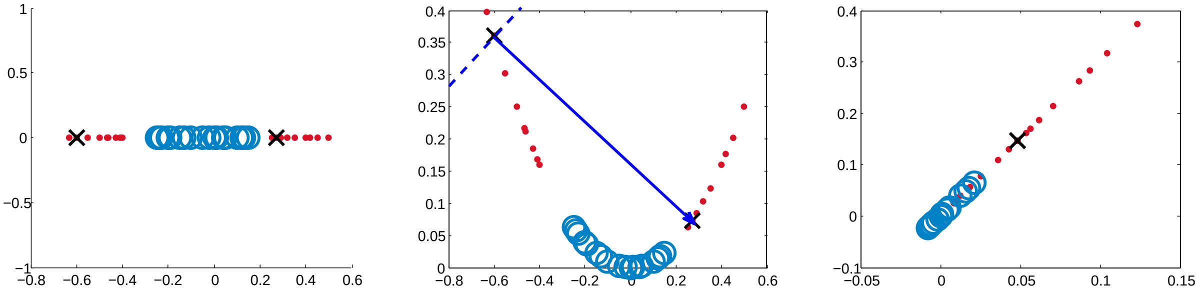
$$d(\mathbf{y}_1, \mathbf{y}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{Z}\mathbf{Z}^T (\mathbf{x}_1 - \mathbf{x}_2)} \qquad (1)$$

which only depends on the original data points and the learned transformation matrix.



(a) (b) (c)

Fig. 1. Illustration of must-link constraints enforcement. (a) Input space. 36 one-dimensional data points originated from two clusters (18 points each, differentiated by markers) that are not linearly separable. Black crosses mark the must-link constraint pair $(\mathbf{m}_1, \mathbf{m}_2)$. (b) The input space is mapped to the 2-dimensional feature space via quadratic mapping $\phi(\mathbf{x}) = [\mathbf{x}\ \mathbf{x}^2]^T$. The blue arrow is the difference vector $(\phi(\mathbf{m}_2) - \phi(\mathbf{m}_1))^T$. The dotted line is the null space. (c) The feature space is projected to the null space of the difference vector. Constrained points collapsed to a single point and a clustering algorithm trivially groups them together.



(a) (b) (c)

Fig. 2. Illustration of a must-link enforcement error on unconstrained data points. Same set-up as Figure 1 with a different pair of must-link constraint. The null space projection result in (c) clearly demonstrates that although the constrained points are mapped to a single point, points from different clusters are mixed together too and leads to clustering mistakes.

B. Integrating Must-link Constraints

Given a pair of must-link constraint $(\mathbf{x}, \mathbf{x}')$, following the idea presented in [8], we can project the input space onto the null space of the difference vector $(\mathbf{x} - \mathbf{x}')^T$, which is the direction orthogonal to the difference vector. Hence, $\mathbf{x}$ and $\mathbf{x}'$ will be mapped to the same point, and the must-link constraint is maximally satisfied. This method does not scale well with the increasing number of must-links. For data with f-dimensional features, if the number of must-link constraints exceeds $f - 1$ all the data points will collapse to a single point. For this reason, we first map data to an enlarged feature space, and then apply the same technique to exploit must-link constraints. We call this method *kernel null space projection*. Figure 1 illustrates this idea using a one-dimensional data set.

Formally, let $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a positive definite kernel function satisfying for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \tag{2}$$

where ϕ is a nonlinear mapping function

$$\phi : \mathcal{X} \mapsto \mathcal{H}$$

that maps input space $\mathcal{X}$ into the f_ϕ-dimensional feature space $\mathcal{H}$. Define the $m \times f_\phi$ *must-link constraint matrix* $\mathbf{M}$ as follows:

$$\mathbf{M} = \begin{bmatrix} (\phi(\mathbf{x}_1) - \phi(\mathbf{x}'_1))^T \\ \vdots \\ (\phi(\mathbf{x}_m) - \phi(\mathbf{x}'_m))^T \end{bmatrix} \tag{3}$$

Then, the projection matrix

$$\mathbf{P} = \mathbf{I}_{f_\phi} - \mathbf{U} \tag{4}$$

where

$$\mathbf{U} = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{\#}\mathbf{M}$$

projects data in $\mathcal{H}$ to the null space of $\mathbf{M}$, and is the desired projection. $\#$ stands for the pseudo-inverse. One can prove that in the null space of $\mathbf{M}$, every pair of must-linked data points collapse to a single point, and thus the must-link constraints are maximally satisfied (see Appendix).

By simple algebra formulation, the projected kernel function is given by

$$\widehat{K}(\mathbf{x},\mathbf{x}') = K(\mathbf{x},\mathbf{x}') - K(\phi(\mathbf{x}),\mathbf{M})^T\mathbf{W}^{\#}K(\phi(\mathbf{x}'),\mathbf{M}) \quad (5)$$

where $K(\phi(\mathbf{x}),\mathbf{M})$ denotes the m-dimensional vector

$$\begin{bmatrix} K(\mathbf{x},\mathbf{x}_1) - K(\mathbf{x},\mathbf{x}'_1) \\ \vdots \\ K(\mathbf{x},\mathbf{x}_m) - K(\mathbf{x},\mathbf{x}'_m) \end{bmatrix}$$

and

$$\mathbf{W}_{i,j} \quad = \quad K(\mathbf{x}_i,\mathbf{x}_j) - K(\mathbf{x}_i,\mathbf{x}'_j) - K(\mathbf{x}'_i,\mathbf{x}_j) + K(\mathbf{x}'_i,\mathbf{x}'_j)$$

Since all the computations of $\widehat{K}(\mathbf{x},\mathbf{x}')$ can be expressed in terms of $K(\mathbf{x},\mathbf{x}')$, the subspace projection is performed implicitly in the kernel space.

Note that, null space projection $\mathbf{P}$ is the optimal projection in the sense that it preserves the variance along the orthogonal directions to the projection direction. Therefore, the original distance measure is best preserved.

C. Integrating Cannot-link Constraints and Data Structure

The kernel null space projection introduced in the last section guarantees the enforcement of must-link constraints by pulling data from the same class close to each other. Thus, the pairwise distances of the embedded data $d(\hat{\phi}(\mathbf{x}),\hat{\phi}(\mathbf{x}'))$ fit the *intra-class* structure better than the pairwise distances in the original space $d(\mathbf{x},\mathbf{x}')$. However, the kernel null space projection can also mistakenly pull data points from different clusters close to each other, thus leading to clustering mistakes. Figure 2 illustrates this issue using the same data as in Figure 1 but with a different pair of must-link constraint. As a result, the pairwise distances of embedded data $d(\hat{\phi}(\mathbf{x}),\hat{\phi}(\mathbf{x}'))$ do not capture the *inter-class* structure well.

This problem can be solved by further exploiting cannot-link constraints based on the kernel null space projection result. The goal of adopting cannot-link constraints is to embed data in a subspace where data points from different classes are further pushed away from each other while the intra-class distance measure is still best preserved. Before presenting how to find such a subspace, let us first make the following declaration and define a few concepts.

Without loss of generality, we assume all the distances have been normalized to $[0,1]$ in our discussion. Then the similarity between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is evaluated as $1 - d(\mathbf{x}_i,\mathbf{x}_j)$. Let $N(\mathbf{x}_i)$ denotes the set of k-nearest neighbors of point $\mathbf{x}_i$ for a given k. Let $\mathbf{S}$ be the *adjacency matrix*, such that

$$\mathbf{S}_{i,j} = \begin{cases} 1 - \hat{d}_\phi(\mathbf{x}_i,\mathbf{x}_j) & \mathbf{x}_i \in N(\mathbf{x}_j) \vee \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0 & otherwise \end{cases} \quad (6)$$

where $\hat{d}_\phi(\mathbf{x}_i,\mathbf{x}_j)$ is the *kernel distance* defined as:

$$\begin{aligned}\hat{d}_\phi(\mathbf{x}_i,\mathbf{x}_j) &= d(\hat{\phi}(\mathbf{x}_i),\hat{\phi}(\mathbf{x}_j)) \\ &= \sqrt{\widehat{K}(\mathbf{x}_i,\mathbf{x}_i) + \widehat{K}(\mathbf{x}_j,\mathbf{x}_j) - 2\widehat{K}(\mathbf{x}_i,\mathbf{x}_j)} \quad (7)\end{aligned}$$

and satisfies $\hat{d}_\phi(\mathbf{x}_i,\mathbf{x}_j) = 0$, if $(\mathbf{x}_i,\mathbf{x}_j) \in \Omega_M$. We adopt the kernel distances in the adjacency matrix because they fit the intra-class structure better.

Let $N(\mathbf{x}_i)^\perp$ be the set of k points that are farthest from $\mathbf{x}_i$ for a given k. In consequence, points in $N(\mathbf{x}_i)^\perp$ tend to originate from a different class than $\mathbf{x}_i$. Let $\mathbf{R}$ be a matrix which is called the *disjoint matrix*, such that

$$\mathbf{R}_{i,j} = \begin{cases} 1 - d(\mathbf{x}_i,\mathbf{x}_j) & \mathbf{x}_i \in N(\mathbf{x}_j)^\perp \vee \mathbf{x}_j \in N(\mathbf{x}_i)^\perp \\ 0 & otherwise \end{cases} \quad (8)$$

Because the disjoint matrix mostly encodes the inter-class structure, the distance measure of the original input space preserves the structure better.

Let $\mathbf{Z} = \begin{bmatrix}\mathbf{z}_1 & \cdots & \mathbf{z}_r\end{bmatrix}$ be the matrix containing r transformation vectors $\mathbf{z}_i|_{i=1}^r$ that embeds data points in the f-dimensional input space in the r-dimensional subspace by $\mathbf{y}_i = \mathbf{Z}^T\mathbf{x}_i$, $\mathbf{x}_i \in \mathbb{R}^f$, $\mathbf{y}_i \in \mathbb{R}^r$. In order to preserve both the intra and inter-class structures, we minimize the following objective function

$$\min \frac{\sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)^2\mathbf{S}_{i,j}}{\sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)^2\mathbf{R}_{i,j}} \quad (9)$$

The numerator incurs heavy penalties if nearby data points (i.e. $\mathbf{S}_{i,j}$ is big) are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then $\mathbf{y}_i$ and $\mathbf{y}_j$ are close as well. The denominator assigns big rewards if nearby data points from different classes (i.e. $\mathbf{R}_{i,j}$ is big) are mapped far away. Therefore, maximizing the denominator has the effect of pushing different classes farther away. Overall, minimizing Eq. (9) both preserves the structure of data and makes the structure more evident.

Similarly, the goal of pushing apart cannot-linked data points is achieved by maximizing the following objective function

$$\max \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\Omega_C} (\mathbf{y}_i - \mathbf{y}_j)^2(1 - d(\mathbf{x}_i,\mathbf{x}_j)) \quad (10)$$

If we modify the disjoint matrix $\mathbf{R}$ to incorporate cannot-link constraints as

$$\widetilde{\mathbf{R}}_{i,j} = \begin{cases} 1 - d(\mathbf{x}_i, \mathbf{x}_j) & \mathbf{x}_i \in N(\mathbf{x}_j)^\perp \vee \mathbf{x}_j \in N(\mathbf{x}_i)^\perp \\ & \vee (\mathbf{x}_i, \mathbf{x}_j) \in \Omega_c \\ 0 & otherwise \end{cases} \tag{11}$$

then the two objectives in Eq. (9) and Eq. (10) can be integrated into a single optimization problem as

$$\begin{aligned} \mathbf{z}^* &= \arg\min_{\mathbf{z}} \frac{\sum_{i,j} (\mathbf{z}^T \mathbf{x}_i - \mathbf{z}^T \mathbf{x}_j)^2 \mathbf{S}_{i,j}}{\sum_{i,j} (\mathbf{z}^T \mathbf{x}_i - \mathbf{z}^T \mathbf{x}_j)^2 \widetilde{\mathbf{R}}_{i,j}} \\ &= \arg\min_{\mathbf{z}} \frac{\mathbf{z}^T \mathbf{X} \mathbf{L}_{\mathbf{S}} \mathbf{X}^T \mathbf{z}}{\mathbf{z}^T \mathbf{X} \mathbf{L}_{\widetilde{\mathbf{R}}} \mathbf{X}^T \mathbf{z}} \end{aligned} \tag{12}$$

where $\mathbf{L}_{\mathbf{S}} = \mathbf{D}^{\mathbf{S}} - \mathbf{S}$ and $\mathbf{L}_{\widetilde{\mathbf{R}}} = \mathbf{D}^{\widetilde{\mathbf{R}}} - \widetilde{\mathbf{R}}$ are the graph Laplacians [4] related to the adjacency matrix $\mathbf{S}$ and the disjoint matrix $\widetilde{\mathbf{R}}$ respectively, and $\mathbf{D}^{\mathbf{S}}$ and $\mathbf{D}^{\widetilde{\mathbf{R}}}$ are diagonal matrices with $\mathbf{D}^{\mathbf{S}}_{i,i} = \sum_j \mathbf{S}_{i,j}$ and $\mathbf{D}^{\widetilde{\mathbf{R}}}_{i,i} = \sum_j \widetilde{\mathbf{R}}_{i,j}$. The r optimal transformation vectors $\mathbf{z}^*_i|_{i=1}^r$ can be found by solving the general eigenvalue problem

$$\mathbf{X} \mathbf{L}_{\mathbf{S}} \mathbf{X}^T \mathbf{z} = \lambda \mathbf{X} \mathbf{L}_{\widetilde{\mathbf{R}}} \mathbf{X}^T \mathbf{z} \tag{13}$$

The r eigen vectors related to the r smallest eigen values are the solution.

Obviously, the performance of the above optimization problem strongly depends on the pairwise distances of data points, which are encoded in matrices $\mathbf{L}_{\mathbf{S}}$ and $\mathbf{L}_{\widetilde{\mathbf{R}}}$. By adopting the kernel distance $\hat{d}_\phi(\mathbf{x}_i, \mathbf{x}_j)$, and distances $d(\mathbf{x}, \mathbf{x}')$ of the original input space, the modification to the feature space in the kernel null space projection step is incorporated. Therefore, the final optimal projection direction is determined by both types of constraints as well as the intrinsic structure of data.

IV. Experiment

A. Datasets

We use multiple real datasets from different domains to evaluate our proposal. Datasets are summarized in Table I. The datasets used are very diverse in terms of size of data, size of feature spaces and number of clusters. In particular, 10 datasets are gathered from the UCI machine learning database [1] because of their popularity in the field of machine learning. Besides, we use the COIL-20 database [2], which is widely used in 3D object recognition research. This database contains gray-scale images of 20 objects. Each object has 72 images taken at different orientations. Thus, the entire database contains 1,440 images. Each image is of size $128 \times 128 = 16,384$ pixels. We further perform bicubic interpolation to downsize every image to 16×16 pixels. This is a commonly used technique to achieve tradeoff between complexity and accuracy. Thus, each image is represented as a vector of dimension 256. Samples of the COIL-20 database are listed in Figure 3.

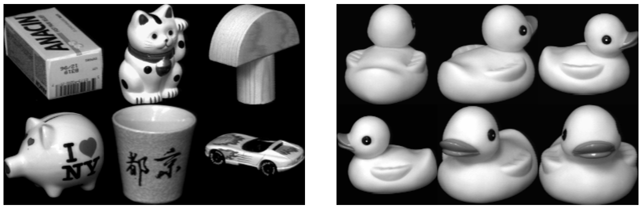


Fig. 3. COIL-20 database. Left: 6 random samples, right: 6 orientations of one object

B. Competitive Techniques and Evaluation

Our proposal has been compared to four state-of-the-art and representative semi-supervised and unsupervised dimensionality reduction techniques. LPPSI [1] is a recent semi-supervised dimensionality reduction method that has been successfully applied to solve face recognition problem. We compare to the kernel version of LPPSI since it is reported to have better performance than the non-kernel version. LPP [5] is an unsupervised dimensionality reduction technique that preserves the local structures of data, and has been widely adopted in visualization and text indexing. SLPP is the supervised version of LPP. PCA is the classical unsupervised dimensionality reduction technique. We test the dimensionality reduction performance achieved by each method in a clustering setting. A better dimensionality reduction technique should reveal the intrinsic structure of the data, and thus leads to higher clustering accuracy. k-means is used as the underlying clustering model for all the experiments. We use the F*-score*, which is a harmonic mean of *precision* and *recall* ranging in $[0, 1]$, to evaluate clustering accuracy. The clustering error rate is defined as $1 - F$-score. All the reported results are based on the average of 20 independent runs.

C. Parameter Setting

For all the kernel methods, we use the RBF kernel, which is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\delta^2}) \tag{14}$$

The parameter δ often significantly influences the performance of kernel methods. With the help of constraints,

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

TABLE I
DATASETS SUMMARY (n: # SAMPLES; f: # FEATURES; k: # CLUSTERS; δ: KERNEL PARAMETER)

dataset	n	f	k	δ
wine	178	13	3	0.6
vehicle	846	18	4	0.9
iris	150	4	3	0.3
balance	625	4	3	0.7
ionosphere	351	34	2	1
glass	214	9	6	0.3
breast	682	10	2	1
Multiple Features	2,000	649	10	0.2
isolet	7,797	617	26	7
Pendigit	10,992	16	10	46
COIL-20	1,440	16,384	20	0.4

we choose the δ value by a simple grid search. For a given δ, we perform the kernel null space projection only, and cluster the projected data. Since the kernel null space projection guarantees that all must-linked data points will be trivially clustered together, we pick the δ value that achieves the maximal clustering accuracy on cannot-link constraints. Empirical results show that this method works very well even with a few pairs of constraints. The δ values chosen for each dataset are listed in Table I. The number of nearest neighbors used in constructing the adjacency and disjoint matrices is set to 5 and is kept the same for all the methods and all the datasets.

D. Fixed Subspace Dimensions

In this experiment, we test the dimensionality reduction performance on datasets with moderate sizes. The purpose is to learn the best projection direction by using all the available data and evaluate the performance. For each dataset and each cluster, we run the experiments by alternatively generating 5 and 20 random pairs of must-link and cannot-link constraints each based on class labels. This end up with $2 \times k \times 5(20)$ pairs of constraints in total for each dataset, where k is the number of clusters. For easy reference, we refer to them as “5(20) pairs” of constraints hereafter. We fix the subspace dimension to be half of the original dimension. Table II shows the evaluation result. On 5 out of 7 datasets, DSP achieves the best F-scores. For the remaining 2 datasets, DSP still shows satisfactory F-scores. Most importantly, when the number of constraints is small (i.e. the 5 pairs case), the performance of DSP is still robust and is better than or similar to the performances of the two unsupervised method PCA and LPP. This means that DSP does not suffer from overfitting, unlike competing methods.

E. Various Subspace Dimensions

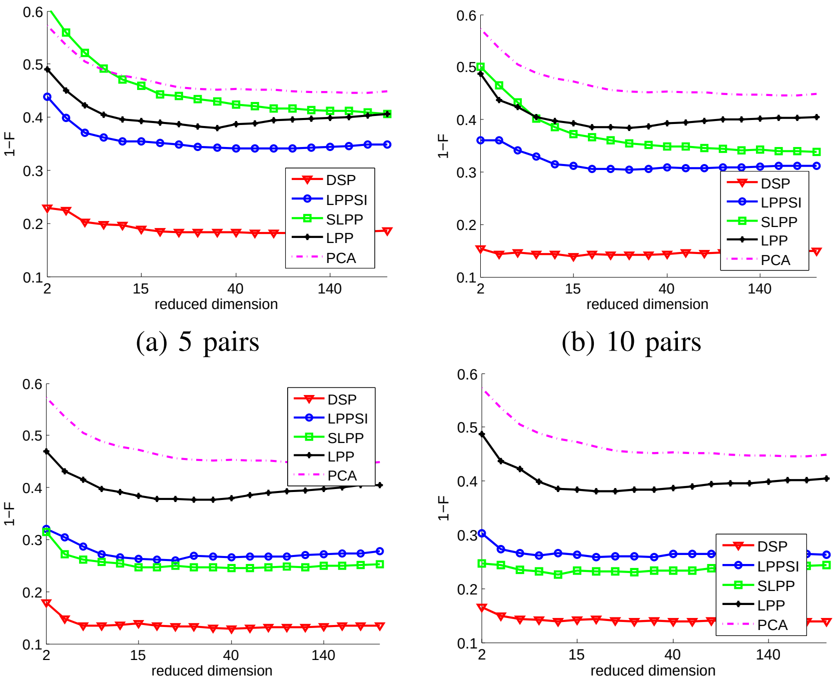
In this experiment, we evaluate the dimensionality reduction techniques for various subspace dimensions. Due to limited space, we show the results on the COIL-20 database for 3D object recognition and the Multiple Features dataset for handwritten digit recognition only, in Figures 4 and 5 respectively. For each dataset 5/10/20/30 pairs of constraints per cluster are generated following the last experiment. The reduced dimensions range from 2 to 200. DSP significantly outperforms other dimensionality reduction techniques for both datasets under all experiment settings. The stable performance of DSP given a few constraints and very low subspace dimensionality is particularly impressive. It is interesting to notice that although LPPSI and SLPP perform well for the COIL-20 dataset, their performances on the digit dataset are worse than the unsupervised LPP for low dimensions and small number of constraint pairs. This effect could be the result of overfitting due to few training data.

F. Generalization

In this experiment, we evaluate how well DSP handles new data points on four large scale datasets. For each dataset, we do 5-fold cross validation. Four folds of data are used for training, which includes generating 20 pairs of constraints and learning the best subspace embedding. Then the one fold testing data points are projected to the learned subspace for further clustering evaluation. Table III shows the generalization performance, compared to the clustering result of testing data without dimensionality reduction. Because the subspace dimensions are significantly smaller than the dimensions of the full feature space, clustering in the subspace will most of the time sacrifice accuracy for efficiency. With

TABLE II
F-SCORE ON HALF-SIZE FEATURE SPACES

	unsupervised		20 pairs			5 pairs		
	PCA	LPP	SLPP	LPPSI	DSP	SLPP	LPPSI	DSP
wine	0.9415	0.9541	0.9563	0.8198	**0.9588**	0.5962	0.7381	**0.9322**
vehicle	0.3070	0.3383	0.6024	0.4092	**0.6042**	0.3417	0.3306	**0.3604**
iris	0.8112	0.7716	0.8920	0.6982	**0.9498**	0.8471	0.6244	**0.9405**
balance	0.5075	0.4754	0.5789	0.5800	**0.6068**	0.5749	**0.5845**	0.5693
ionoshpere	0.6050	0.6050	0.7061	0.6205	**0.7211**	0.6108	0.5992	**0.7145**
glass	0.3950	0.3903	**0.4032**	0.4023	0.3833	0.3849	0.3058	**0.4131**
breast	0.9307	0.9307	0.9027	**0.9352**	0.9202	0.7478	**0.9292**	0.9288



(a) 5 pairs

(b) 10 pairs

(c) 20 pairs

(d) 30 pairs

Fig. 4. Error Rate vs. Reduced Dimensions for 3D object recognition

TABLE III
F-SCORE FOR GENERALIZATION (r: SUBSPACE DIMENSIONALITY)

	full feature	DSP-generalize(r)
Multiple Features	0.7101	0.9459(20)
isolet	0.5311	0.4740(20)
Pendigit	0.5502	0.5873(5)
COIL-20	0.5732	0.7872(20)

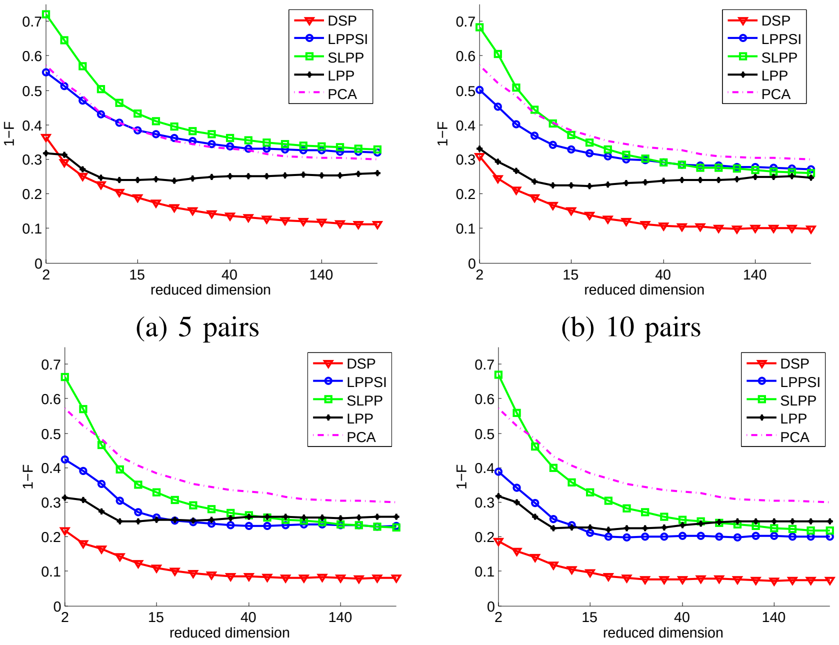
the help of constraints, for three out of four datasets, the clustering accuracy after DSP reduction is in fact being improved. This indicates that DSP is effective in exploiting constraints and generalizing to new data points.

V. Conclusion

We propose a novel semi-supervised dimensionality reduction technique based on subspace projections in both the kernel space and the original input space. Projections in the two spaces interact and data are embedded in an optimal low-dimensional subspace where the intrinsic structure of data is more evident, and thus eases the subsequent data analysis. Experiments on multiple real datasets clearly demonstrate that significant improvement in learning accuracy can be achieved after our dimensionality reduction is employed with only a few constraints.

VI. Appendix

We prove that in the null space of $\mathbf{M}$, every pair of must-linked data points collapse to a single point, and



(a) 5 pairs

(b) 10 pairs

(c) 20 pairs

(d) 30 pairs

Fig. 5. Error Rate vs. Reduced Dimensions for handwritten digit recognition

thus the must-link constraints are maximally satisfied.

PROOF. Let $(\phi(\mathbf{x}_i), \phi(\mathbf{x'}_i))$ be the i-th pair of must-link data points in the kernel space $\mathcal{H}$. For any data point $\phi(\mathbf{x}) \in \mathcal{H}$, its embedding in the null space of $\mathbf{M}$ is given by:

$$\hat{\phi}(\mathbf{x}) = \mathbf{P}\phi(\mathbf{x}) \tag{15}$$

Given $\mathbf{P}$ as defined in Eq. (4), we then have

$$\hat{\phi}(\mathbf{x}_i) - \hat{\phi}(\mathbf{x}'_i) = \mathbf{P}(\phi(\mathbf{x}_i) - \phi(\mathbf{x}'_i))$$

$$\begin{aligned} &= (\mathbf{I} - \mathbf{U})(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) - \mathbf{U}(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) - (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= 0 \end{aligned} \tag{16}$$

The identity $\mathbf{U}(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) = (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i))$ follows from the fact that $(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i))$ is in the row space of $\mathbf{M}$. Since $\mathbf{P}$ is not null, we get

$$\hat{\phi}(\mathbf{x}_i) = \hat{\phi}(\mathbf{x'}_i) \tag{17}$$

Thus the two points are mapped to the same point.(q.e.d)

REFERENCES

[1] S. An, W. Liu, and S. Venkatesh. Exploiting side information in locality preserving projection. In *CVPR*, pages 1–8, 2008.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *ICML*, pages 11–18, 2003.

[3] H. Cevikalp, J. Verbeek, F. Jurie, and A. Kläser. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, pages 489–496, 2008.

[4] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[5] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[6] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, pages 2072–2078, 2006.

[7] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 12 2000.

[8] O. Tuzel, F. Porikli, and P. Meer. Kernel methods for weakly supervised mean shift clustering. In *ICCV*, pages 59–68, 2009.

[9] M. Verleysen. Learning high-dimensional data. In *Limitations and Future Trends in Neural Computation*, pages 141–162, 2003.

[10] D. Zhang, Z.-H. Zhou, and S. Chen. Semi-supervised dimensionality reduction. In *SDM*, 2007.