Neurocomputing 76 (2012) 114–124

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semi-supervised dimensionality reduction for analyzing high-dimensional data with constraints ☆

Su Yan [a,*], Sofien Bouaziz [b], Dongwon Lee [c], Jesse Barlow [c]

[a] IBM Almaden Research Center, 650 Harry Rd, San Jose, CA 95120, USA
[b] Ecole Polytechnique Fédérale de Lausanne, Switzerland
[c] The Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Available online 3 August 2011

Keywords:
Dimensionality reduction
Semi-supervised learning
Clustering
Kernel methods
Linear transformation
Generalized eigenproblem

ABSTRACT

In this paper, we present a novel semi-supervised dimensionality reduction technique to address the problems of inefficient learning and costly computation in coping with high-dimensional data. Our method named the dual subspace projections (DSP) embeds high-dimensional data in an optimal low-dimensional space, which is learned with a few user-supplied constraints and the structure of input data. The method projects data into two different subspaces respectively the kernel space and the original input space. Each projection is designed to enforce one type of constraints and projections in the two subspaces interact with each other to satisfy constraints maximally and preserve the intrinsic data structure. Compared to existing techniques, our method has the following advantages: (1) it benefits from constraints even when only a few are available; (2) it is robust and free from overfitting; and (3) it handles nonlinearly separable data, but learns a linear data transformation. As a conclusion, our method can be easily generalized to new data points and is efficient in dealing with large datasets. An empirical study using real data validates our claims so that significant improvements in learning accuracy can be obtained after the DSP-based dimensionality reduction is applied to high-dimensional data.

1. Introduction

High-dimensional data are prevalent in a wide variety of areas and have become a significant challenge for data mining, archiving, indexing and downstream analysis. 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 [12]. Second, the computational cost is so high that many techniques are not readily applicable to handle large amount of high-dimensional data [28].

Dimensionality reduction is a technique that helps solving the high dimensionality problem and has been extensively studied and widely applied in text analysis [10], face recognition [3], and microarray gene expression analysis [8] where data are usually expressed as vectors of high dimension. Dimensionality reduction is also the key technique for data compression that enables efficient information storage and retrieval [15], as well as for data visualization, where high-dimensional data are mapped to 2D or 3D spaces helping the user gain a qualitative understanding of the information [24].

A dimensionality reduction technique finds low-dimensional structures of data hidden in high-dimensional observations. *Feature selection*[16] and *feature reduction*[18,17] are 2D reduction solutions. Feature selection reduces dimensionality by selecting a subset of existing features. Thus, the physical interpretation of each feature is preserved in the reduced space. However, in removing many features prior to learning from the data, information about the underlying data may be lost. Feature reduction reduces dimensionality by combining features with linear or nonlinear transformations. A feature reduction approach can greatly reduce the feature space dimensionality while still preserve discriminative information, although it does not retain feature's physical interpretations. In general, the choice between feature reduction and feature selection depends on the application domain. In this work, we focus on the feature reduction technique because for many web-related data analysis problems (such as web data clustering, indexing, collaborative filtering etc.), efficient data representation that preserves discriminative information is more critical for fast similarity comparison and search.

☆This paper is invited as an extension of authors' paper: *DSP: Robust Semi-Supervised Dimensionality Reduction Using Dual Subspace Projections*, Su Yan, Sofien Bouaziz, and Dongwon Lee, 2010 IEEE/WIC/ACM International Conferences.

* Corresponding author.
E-mail addresses: sueyan@gmail.com, syan@us.ibm.com (S. Yan), sofien.bouaziz@epfl.ch (S. Bouaziz), dongwon@psu.edu (D. Lee), barlow@cse.psu.edu (J. Barlow).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.03.057

Recently, *semi-supervised dimensionality reduction* has stirred many research interests [2,14,1,13,25,30,5,6]. 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.

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. Compared to supervision in the form of labeled data, constraints are more general, because labeled data can be represented in the form of pairwise constraints but the inverse does not hold. The goal of semi-supervised dimensionality reduction is to embed high-dimensional data into a lower dimensional subspace with the help of pairwise constraints. If the dimensionality reduction process can indeed benefit from constraints, the data embedded in the subspace will show more evident clustering structure than without using constraints. For this reason, the performance of semi-supervised dimensionality reduction can be measured by the clustering performance achieved on the embedded low-dimensional data.

To incorporate constraints in dimensionality reduction, [2] introduces relevant component analysis (RCA) that exploits must-links only. Ref. [14] introduces discriminant component analysis (DCA) that extends RCA by also exploring cannot-links. Recently, [1] proposes to incorporate constraints using a modified locality preserving projection (LPP) [13] 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, [30,5] propose semi-supervised dimensionality reduction methods that exploit both constraints and the structure of 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 nonlinearly separable data. However, because the projection is done implicitly in the kernel space, the transformation matrix that maps high-dimensional data to low dimensions is not explicitly learned. The mappings are defined only on the training data points. As a result, these methods do not generalize well to new data points. To be specific, in order to compute the projection of test points all the training points need to be stored, and the inner product between the test points to all the training points needs to be calculated and stored. 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, the 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, yet the learned mapping is still linear. Therefore, generalizing to test data is efficient.

2. Background and related work

Dimensionality reduction is the technique that extracts low-dimensional structure in high-dimensional data. The algorithms for dimensionality reduction can be broadly categorized as *global* vs. *local* methods, and *linear* vs. *nonlinear* methods.

2.1. Global vs. local

Representative global methods include principal component analysis (PCA), multidimensional scaling (MDS), and linear discriminant analysis (LDA). PCA is an unsupervised method that maximally preserves the variance of data; classical MDS finds an embedding that preserves the inter-point distances; LDA is a supervised method that achieves maximal class separation by maximizing the ratio of between-class variance to the within-class variance. The principal advantage of global approaches is that they tend to give a more faithful representation of data global structure. However, the local geometry of data maybe lost.

To overcome the drawbacks of global methods and their variants, a number of *local* dimensionality reduction methods have been proposed, such as locality preserving projections (LPP) [13], locally linear embedding (LLE) [20], Laplacian eigenmaps [4]. These methods embed data in a low-dimensional space such that nearby data points in the original space are still near to each other in the embedded space.

Global structures and local structures of a dataset are both important for learning from high-dimensional data. This leads to work that combine the advantages of global approaches with the advantages of local methods to get the best of both worlds [9,23]. The dimensionality reduction approach introduced in this paper falls into this category and preserves both the global and local structures of data.

2.2. Linear vs. nonlinear

Classical dimensionality reduction methods such as PCA, MDS, LDA are linear methods. They are simple to implement, efficiently computable, and perform well in general. However, when sever nonlinearity is involved in data, linear methods are less effective. Nonlinear dimensionality reduction methods, such as ISOMAP [26], LLE, Laplacian eigenmaps, are proposed based on spectral techniques.

Another common and effective solution to the nonlinearity problem is to use the popular kernel technique [21]. Kernel technique is based on the idea that nonlinearly separable data can be separated linearly in some high-dimensional space. Data are first mapped to a high-dimensional feature space by nonlinear transformations, then can be separated in the kernel-space with simple linear methods. Most dimensionality reduction methods have their kernel space equivalents to deal with nonlinearities in data, for example, kernel PCA, kernel LDA, kernel LPP, etc.

Given the ability to handle nonlinear data, nonlinear methods also have certain drawbacks compared to linear methods. For example, the mapping defined by nonlinear spectral methods are defined only on the training data points and it is difficult, if not impossible to evaluate the mapping for new test points. The same observation holds for kernel-based methods. Moreover, kernel machines easily overfit. Given pairwise constraints, it is always possible to find a data partition that satisfies all the constraints in certain high-dimensional space. Therefore, kernel machine will overfit with limited constraints, since the nonlinear mapping that satisfies a few pairs of constrains does not necessarily best reveal the structure in data.

3. 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 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 technique. 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 two data points constrained by a must-link will be mapped to a single point. This idea originates from [27], 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. The proposed method exploits kernel techniques to handle nonlinearly separable data. However, the learned transformation is still linear, and thus can be easily generalized to new data points.

Through the paper, we use the following notation conventions. A matrix is represented by a capitalized boldface letter; a vector is represented by a lowercase boldface letter; a scalar is represented by an italic lowercase letter, and a function is represented by an italic letter. In particular, Table 1 lists major symbols and their meaning.

Table 1
Major symbols and meaning.

Symbol	Meaning
T	Matrix transpose
$\#$	Matrix pseudo-inverse
$\mathbf{I}$	Identity matrix
Ω_M	The set of must-links
Ω_C	The set of cannot-links
$\mathbf{X}$	Matrix of input data
$\mathbf{x}_i$	The ith data point
$\mathbf{Y}$	Matrix of embedded data
$\mathbf{Z}$	Transformation matrix to be learned
$d(\mathbf{x}_i,\mathbf{x}_j)$	Distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the input space
$\hat{d}_{\phi}(\mathbf{x}_i,\mathbf{x}_j)$	Distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ after kernel null space projection
$\phi(\cdot)$	Implicit nonlinear mapping function
$K(\mathbf{x}_i,\mathbf{x}_j)$	Kernel function
$\tilde{K}(\mathbf{x}_i,\mathbf{x}_j)$	Kernel function after kernel null space projection
$\mathbf{M}$	Must-link constraint matrix
$\mathbf{S}$	Adjacency matrix
$\mathbf{R}$	Disjoint matrix
$\tilde{\mathbf{R}}$	Disjoint matrix after incorporating cannot-links

4. Main proposal

4.1. Problem setting

We therefore consider the following problem. Given a high-dimensional dataset $\mathbf{X}=(\mathbf{x}_1,\ldots,\mathbf{x}_n)$ of input patterns where $\mathbf{x}_i\in\mathbb{R}^f$, how can we compute n corresponding output patterns $\mathbf{y}_i\in\mathbb{R}^r$, $r\ll f$, that provide a "faithful" low-dimensional representation?

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

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

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

4.2. Integrating must-link constraints

Given a pair of must-link constraint $(\mathbf{x},\mathbf{x}')$, following the idea presented in [27], 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*. Fig. 1 illustrates this idea using a 1D dataset.

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).

Given the data points and must-link constraints, the kernel null space projection maps the data points in the feature space to

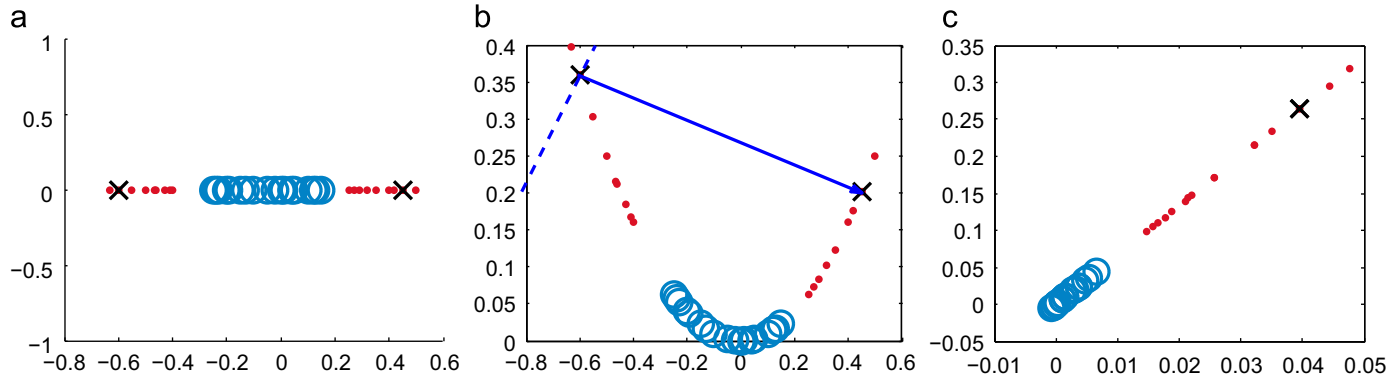

Fig. 1. Illustration of must-link constraints enforcement. (a) Input space. 36 one-dimensional data points originated from two clusters (18 points each, differentiated by markers) that are not linearly separable. Black crosses mark the must-link constraint pair ($\mathbf{m}_1,\mathbf{m}_2$). (b) The input space is mapped to the two-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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

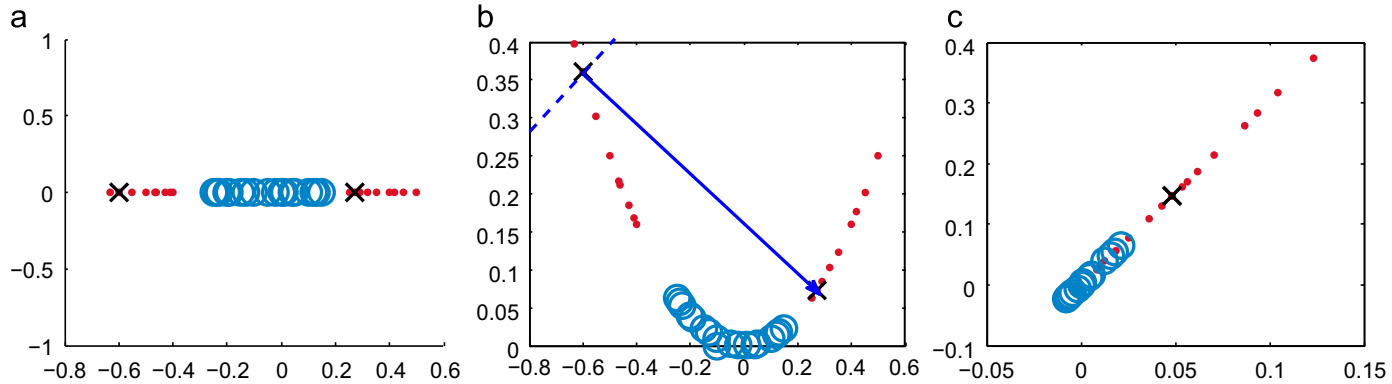

Fig. 2. Illustration of a must-link enforcement error on unconstrained data points. Same set-up as Fig. 1 with a different pair of must-link constraint. The null space projection result in (c) clearly demonstrates that although the constrained points are mapped to a single point, points from different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

the null space of the must-link constraint matrix by

$$\hat{\phi}(\mathbf{x})=\mathbf{P}\phi(\mathbf{x}) \tag{5}$$

Since the implicit nonlinear mapping function $\phi(\cdot)$ is unknown, the projection cannot be performed explicitly. A closer look at the kernel function after the kernel null space projection reveals that the projection can be performed implicitly in the kernel space. That is, the kernel function after subspace projection has the form

$$\begin{aligned}\widehat{K}(\mathbf{x},\mathbf{x}')&=\hat{\phi}(\mathbf{x})^T\hat{\phi}(\mathbf{x}')=\phi(\mathbf{x})^T\mathbf{P}^T\mathbf{P}\phi(\mathbf{x}')=\phi(\mathbf{x})^T\mathbf{P}\phi(\mathbf{x}')\\&=\phi(\mathbf{x})^T(\mathbf{I}-\mathbf{M}^T\mathbf{W}^{\#}\mathbf{M})\phi(\mathbf{x}')\\&=K(\mathbf{x},\mathbf{x}')-K(\phi(\mathbf{x}),\mathbf{M})^T\mathbf{W}^{\#}K(\phi(\mathbf{x}'),\mathbf{M})\end{aligned} \tag{6}$$

The identity $\mathbf{P}^T\mathbf{P}=\mathbf{P}$ follows from the fact that $\mathbf{P}$ is a projection matrix. $K(\phi(\mathbf{x}),\mathbf{M})$ denotes the m-dimensional vector

$$\begin{bmatrix}K(\mathbf{x},\mathbf{x}_1)-K(\mathbf{x},\mathbf{x}'_1)\\ \vdots \\ K(\mathbf{x},\mathbf{x}_m)-K(\mathbf{x},\mathbf{x}'_m)\end{bmatrix} \tag{7}$$

and

$$\mathbf{W}_{i,j}=K(\mathbf{x}_i,\mathbf{x}_j)-K(\mathbf{x}_i,\mathbf{x}'_j)-K(\mathbf{x}'_i,\mathbf{x}_j)+K(\mathbf{x}'_i,\mathbf{x}'_j) \tag{8}$$

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, the 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.

4.3. 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. Fig. 2 illustrates this issue using the same data as in Fig. 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

$$\mathbf{S}_{i,j}=\begin{cases}1-\hat{d}_{\phi}(\mathbf{x}_i,\mathbf{x}_j) & \mathbf{x}_i\in N(\mathbf{x}_j)\vee\mathbf{x}_j\in N(\mathbf{x}_i)\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

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

$$\hat{d}_{\phi}(\mathbf{x}_i,\mathbf{x}_j)=d(\hat{\phi}(\mathbf{x}_i),\hat{\phi}(\mathbf{x}_j))=\sqrt{\widehat{K}(\mathbf{x}_i,\mathbf{x}_i)+\widehat{K}(\mathbf{x}_j,\mathbf{x}_j)-2\widehat{K}(\mathbf{x}_i,\mathbf{x}_j)} \tag{10}$$

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

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

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

Let $\mathbf{Z}=[\mathbf{z}_1\ \cdots\ \mathbf{z}_r]$ be the matrix containing r transformation vectors $\mathbf{z}_i|_{i=1}^{r}$ that embed 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

$$\min\frac{\sum_{i,j}(\mathbf{y}_i-\mathbf{y}_j)^2\mathbf{S}_{i,j}}{\sum_{i,j}(\mathbf{y}_i-\mathbf{y}_j)^2\mathbf{R}_{i,j}} \tag{12}$$

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. (12) 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

$$\max\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\Omega_C}(\mathbf{y}_i-\mathbf{y}_j)^2(1-d(\mathbf{x}_i,\mathbf{x}_j)) \tag{13}$$

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

$$\tilde{\mathbf{R}}_{i,j}=\begin{cases}1-d(\mathbf{x}_i,\mathbf{x}_j) & \mathbf{x}_i\in N(\mathbf{x}_j)^{\perp}\vee\mathbf{x}_j\in N(\mathbf{x}_i)^{\perp}\\ & \vee\ (\mathbf{x}_i,\mathbf{x}_j)\in\Omega_c\\ 0 & \text{otherwise}\end{cases} \tag{14}$$

then the two objectives in Eqs. (12) and (13) can be integrated into a single optimization problem as

$$\mathbf{z}^*=\arg\min_{\mathbf{z}}\frac{\sum_{i,j}(\mathbf{z}^T\mathbf{x}_i-\mathbf{z}^T\mathbf{x}_j)^2\mathbf{S}_{i,j}}{\sum_{i,j}(\mathbf{z}^T\mathbf{x}_i-\mathbf{z}^T\mathbf{x}_j)^2\tilde{\mathbf{R}}_{i,j}}=\arg\min_{\mathbf{z}}\frac{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T\mathbf{z}}{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T\mathbf{z}} \tag{15}$$

where $\mathbf{L}_{\mathbf{S}}=\mathbf{D}^{\mathbf{S}}-\mathbf{S}$ and $\mathbf{L}_{\tilde{\mathbf{R}}}=\mathbf{D}^{\tilde{\mathbf{R}}}-\tilde{\mathbf{R}}$ are the graph Laplacians [7] related to the adjacency matrix $\mathbf{S}$ and the disjoint matrix $\tilde{\mathbf{R}}$ respectively, and $\mathbf{D}^{\mathbf{S}}$ and $\mathbf{D}^{\tilde{\mathbf{R}}}$ are diagonal matrices with $\mathbf{D}^{\mathbf{S}}_{i,i}=\sum_j\mathbf{S}_{i,j}$ and $\mathbf{D}^{\tilde{\mathbf{R}}}_{i,i}=\sum_j\tilde{\mathbf{R}}_{i,j}$. For dimensionality reduction, we pick the r optimal transformation vectors $\mathbf{z}_i^*|_{i=1}^{r}$ to compose the transformation matrix $\mathbf{Z}\in\mathbb{R}^{f\times r}$.

Note that Eq. (15) is the expression of the *generalized Rayleigh quotient* [11]. The solutions can be found by solving a generalized eigenvalue problem. To see this, let us denote

$$g(\mathbf{z})=\frac{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T\mathbf{z}}{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T\mathbf{z}}=\frac{\mathbf{z}^T\mathbf{A}\mathbf{z}}{\mathbf{z}^T\mathbf{B}\mathbf{z}} \tag{16}$$

where

$$A=\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T,\quad B=\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T \tag{17}$$

The construction of matrices $\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T$ and $\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T$ ensures both are symmetric and positive semi-definite. We determine the extremum points of $g(\mathbf{z})$, i.e., the points $\mathbf{z}^*$ satisfying $\nabla g(\mathbf{z})=\mathbf{0}$. The gradient $\nabla g(\mathbf{z})$ is calculated as

$$\nabla g(\mathbf{z})=\frac{2\mathbf{A}\mathbf{z}(\mathbf{z}^T\mathbf{B}\mathbf{z})-2(\mathbf{z}^T\mathbf{A}\mathbf{z})\mathbf{B}\mathbf{z}}{(\mathbf{z}^T\mathbf{B}\mathbf{z})^2} \tag{18}$$

$$\nabla g(\mathbf{z})=\frac{2\mathbf{A}\mathbf{z}-2\,g(\mathbf{z})\mathbf{B}\mathbf{z}}{\mathbf{z}^T\mathbf{B}\mathbf{z}} \tag{19}$$

By setting $\nabla g(\mathbf{z})=0$, we have

$$\mathbf{A}\mathbf{z}=g(\mathbf{z})\mathbf{B}\mathbf{z} \tag{20}$$

which is the form of the generalized eigenvalue problem. Thus, the extremum points $\mathbf{z}^*$ (with the corresponding extreme values $g(\mathbf{z}^*)$) of Eq. (15) are obtained as the eigenvectors (eigenvalues) of the corresponding generalized eigenproblem

$$\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T\mathbf{z}=\lambda\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T\mathbf{z} \tag{21}$$

In particular, the r eigenvectors related to the r smallest nonzero eigenvalues 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 $\mathbf{L}_{\tilde{\mathbf{R}}}$. 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.

The details of the DSP method is listed in Algorithm 1.

Algorithm 1. Dual Subspace Projections (DSP)

Input

a set of n data points in f dimensions: $\{\mathbf{x}_i\}_{i=1}^{n}\in\mathcal{X}$;
two sets of constraints:
must-links: $\Omega_M=\{(\mathbf{x}_i,\mathbf{x}_i')\}_{i=1}^{m}$;
cannot-links: $\Omega_C=\{(\mathbf{x}_i,\mathbf{x}_i')\}_{i=1}^{c}$;
desired subspace dimension: r;

Output

optimal transformation matrix $\mathbf{Z}\in\mathbb{R}^{f\times r}$ that maps f-dimensional features to r-dimensional subspace;
embedded data points: $\mathbf{y}=\mathbf{Z}^T\mathbf{x}$,

Kernel null space projection:
Compute $K(\mathbf{x},\mathbf{x}')$ by Eq. (2);
Compute $K(\phi(\mathbf{x}),\mathbf{M})$ by Eq. (7);
Compute $\mathbf{W}$ by Eq. (8);
Compute $\widehat{K}(\mathbf{x},\mathbf{x}')$ by Eq. (6);
Dual subspace projection:
Compute kernel distances by Eq. (10)
Normalize the kernel distances
Compute $\mathbf{S}$ by Eq. (9);
Compute $\tilde{\mathbf{R}}$ by Eq. (14) ;
Compute $\mathbf{Z}^*$ by solving GeneralizedEigenProblem($\mathbf{S}$, $\tilde{\mathbf{R}}$, $\mathbf{X}$, r) (Eq. (21));
Embed input data by $\mathbf{Y}=\mathbf{Z}^T\mathbf{X}$.
Transformation matrix $\mathbf{Z}$;
embedded data $\mathbf{Y}$.

4.4. Computation

Like many other dimensionality reduction algorithms (i.e. LPP, LDA etc.), the DSP method is formulated as a generalized eigenvalue problem, as shown in Eq. (20). Well-established algorithms in numerical linear algebra have been developed. The generalized eigenvalue problems usually take $O(n^3)$ flops (1 "flop" corresponds to 1 floating-point multiplication and addition), where n is the number of data points in DSP method. Depending on the special structure of the matrices $\mathbf{A}$ and $\mathbf{B}$, the order constant varies in a large range. The DSP problem is solved as a *symmetric positive definite generalized eigenvalue problem* [22], which has a modest order constant.

For our DSP problem, both $\mathbf{A}$ and $\mathbf{B}$ are positive semi-definite. The *simultaneous diagonalization* algorithm [19,29] is adopted to solve the DSP problem efficiently. In particular, we starts the algorithm by diagonalizing $\mathbf{B}$. If $\mathbf{B}$ is nonsingular, the algorithm proceeds with no modifications. If $\mathbf{B}$ is singular, after $\mathbf{B}$ is diagonalized, the 0 (or close to 0) eigenvalues and corresponding eigenvectors should be discarded for the algorithm to proceed. The effect of this step is to discard the null space of $\mathbf{B}$[1] to make it nonsingular. The reason to discard the null space of $\mathbf{B}$ is that the null space of $B=\mathbf{X}\mathbf{L}_{\tilde{\mathbf{R}}}\mathbf{X}^T$ carries no discriminative information, but the null space of $A=\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T$ carries most of the discriminative information, and therefore should be preserved. To see this, for a projection direction $\mathbf{z}$, if $\mathbf{A}\mathbf{z}=0$, and $\mathbf{B}\mathbf{z}\neq 0$, Eq. (16) is minimized since all solutions are non-negative. Detailed explanation to the simultaneous digitalization algorithm for singular matrices can be found in [29].

Moreover, in DSP, the transformation matrix $\mathbf{Z}$ is learned offline. During the online reduction phrase, the number of data points n does not impact the performance of dimensionality reduction.

4.5. Subspace kernel K-means (SKK-means)

In this section, we present a novel semi-supervised clustering method, subspace kernel K-means (SKK-means), which incorporates must-link constraints in the kernel space by adopting the kernel null space projection method introduced in Section 4.2. The objective of presenting SKK-means is twofold. First, the performance of SKK-means can validate the effectiveness of kernel null space projection in exploring supervision in the form of must-links. Second, since DSP further explores cannot-links as well as the intrinsic structure of the input data on top of kernel null space projection, comparing DSP to SKK-means allows us to check whether or not dual subspace projections provide a better data representation than kernel null space projection alone.

SKK-means is a simple extension to kernel K-means, which enhances the standard K-means clustering algorithm to identify nonlinearly separable clusters by the use of a kernel function. Let π_i denote the ith cluster of the total k clusters, and a partitioning of data points be $\{\pi_i\}_{i=1}^k$. Kernel K-means generates a data partition by minimizing the following objective function

$$D(\{\pi_i\}_{i=1}^k)=\sum_{i=1}^{k}\sum_{\mathbf{x}\in\pi_i}\|\phi(\mathbf{x})-\mathbf{m}_i\|^2 \tag{22}$$

where

$$\mathbf{m}_i=\frac{1}{|\pi_i|}\sum_{\mathbf{t}\in\pi_i}\phi(\mathbf{t}) \tag{23}$$

represents the centroid of cluster π_i in the kernel space, and $|\pi_i|$ is the size of cluster π_i. With the same idea of the standard K-means, kernel K-means assigns a data point to the nearest cluster, where the point to cluster distance is measured as the point to cluster centroid distance in the kernel space.

To assign a data point to a cluster at each iteration, the Euclidean distance from the point $\phi(\mathbf{x})$ to centroid $\mathbf{m}_i$ needs to be calculated and is given by

$$\|\phi(\mathbf{x})-\mathbf{m}_i\|^2=K(\mathbf{x},\mathbf{x})-\frac{2}{|\pi_i|}\sum_{\mathbf{t}\in\pi_i}K(\mathbf{x},\mathbf{t})+\frac{1}{|\pi_i|^2}\sum_{\mathbf{t},\mathbf{t}'\in\pi_i}K(\mathbf{t},\mathbf{t}')$$

The evaluation of the right-hand side of the above equation only involves the kernel function $K(\cdot,\cdot)$ and the input data points, and thus can be solved in the kernel space.

In SKK-means, in order to incorporate must-links, the objective is to assign a data point after kernel null space projection $\hat{\phi}(\mathbf{x})$ to the nearest cluster, and the cluster centroid $\hat{\mathbf{m}}_i$ is also evaluated in the projected subspace. By simple algebra formulation, the SKK-means objective function is given by

$$D(\{\pi_i\}_{i=1}^k)\big|_{SKK\_means}=\sum_{i=1}^{k}\sum_{\mathbf{x}\in\pi_i}\|\hat{\phi}(\mathbf{x})-\hat{\mathbf{m}}_i\|^2 \tag{24}$$

and the Euclidean distance from a point to a cluster centroid after the kernel null space projection has the following form

$$\|\hat{\phi}(\mathbf{x})-\hat{\mathbf{m}}_i\|^2=\widehat{K}(\mathbf{x},\mathbf{x})-\frac{2}{|\pi_i|}\sum_{\mathbf{t}\in\pi_i}\widehat{K}(\mathbf{x},\mathbf{t})+\frac{1}{|\pi_i|^2}\sum_{\mathbf{t},\mathbf{t}'\in\pi_i}\widehat{K}(\mathbf{t},\mathbf{t}') \tag{25}$$

where $\widehat{K}(\cdot,\cdot)$ is given by Eq. (6), and $\hat{\mathbf{m}}_i$ denotes the cluster centroid of cluster π_i in subspace. As the above equation shows, the evaluation of the point to cluster distance only involves $\widehat{K}(\cdot,\cdot)$ which is the kernel function after null space projection and the input data points, and thus can be solved implicitly in the kernel space.

5. Experiment

5.1. Datasets

We use multiple real datasets from different domains to evaluate our proposal. Datasets are summarized in Table 2. The datasets used are very diverse in terms of size of data, size of feature space and number of clusters. In particular, 11 datasets are gathered from the UCI machine learning database[2] because of their popularity in the field of machine learning. We use the COIL-20 database,[3] 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 1440 images. Each image is of size $128\times128=16{,}384$ 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 Fig. 3. Moreover, we evaluate our proposals with the KDD-Cup-99[4] data for intrusion detector learning. Detailed processing of this dataset is introduced later for easy reference.

5.2. Competitive techniques

The standard K-means clustering method is adopted as the baseline. K-means results show that the clustering performance one can achieve in the original input feature space without

[1] Null space of $\mathbf{B}=\{\mathbf{x}|\mathbf{B}\mathbf{x}=0,\mathbf{x}\in\mathbb{R}^n\}$.

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

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

[4] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Table 2
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
Sonar	208	60	2	0.8
Multiple features	2000	649	10	0.2
Isolet	7797	617	26	7
Pendigit	10,992	16	10	46
COIL-20	1440	16,384	20	0.4
Intrusion detection	494,021	41	23	0.7

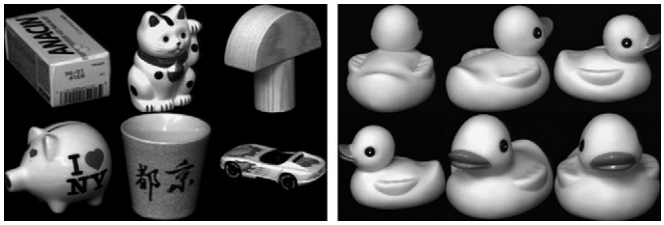

Fig. 3. COIL-20 database. Left: six random samples; right: six orientations of one object.

dimensionality reduction and without exploring supervision in the form of constraints. SKK-means is evaluated to show how effective the kernel null space projection in exploring must-links. Our main proposal, DSP, is then evaluated to show its ability of exploring both must-links and cannot-links, as well as the intrinsic structure of the input data to achieve better data representation in a much reduced subspace.

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 [13] 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. The standard K-means is used as the underlying clustering model for all the dimensionality reduction techniques.

5.3. Evaluation

We use two metrics, *F-score* and rand index (*RI*), to evaluate clustering accuracy. Let $\mathbb{T}$ denote the set of pairs of data items that belong to a same cluster according to ground truth and $\mathbb{R}$ denote the set of pairs of data items that have been assigned to a same cluster by the clustering algorithm. Then, *precision* and *recall* are defined as

$$precision = \frac{|\mathbb{R} \cap \mathbb{T}|}{|\mathbb{R}|}$$

$$recall = \frac{|\mathbb{R} \cap \mathbb{T}|}{|\mathbb{T}|}$$

F-score is a harmonic mean of *precision* and *recall* defined as

$$Fscore = \frac{2 \times precision \times recall}{precision + recall} \tag{26}$$

The Rand Index (RI) measures the degree of similarity in terms of pairwise co-assignments between the cluster membership C from the ground truth and the solution $\hat{C}$ generated by a clustering algorithm. It is defined as

$$RI(C,\hat{C}) = \frac{|c_i = c_j \wedge \hat{c}_i = \hat{c}_j| + |c_i \neq c_j \wedge \hat{c}_i \neq \hat{c}_j|}{n(n-1)/2} \tag{27}$$

where c_i and $\hat{c}_i$ are the cluster membership of item i according to C and $\hat{C}$, and n is the number of data items being clustered. Obviously, RI penalizes both the false positive and false negative decisions during clustering.

Both F-score and RI take value in the range of [0,1], where 1 means 100% clustering accuracy. We also define the "clustering error rate" is as 1-F-score. All the reported results are based on the average of 20 independent runs.

5.4. Parameter setting

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

$$K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\delta^2}\right) \tag{28}$$

The parameter δ often significantly influences the performance of kernel methods. With the help of constraints, 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 using SSK-means. 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 2. 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.

5.5. Exploiting constraints

In this experiment, we evaluate how effective the proposed subspace projection method is in exploring different types of constraints and data structures. The real-world intrusion detection data of KDD-Cup-99 is used for this experiment. The dataset was originally provided for supervised classification tasks. We use the provided labels of training data as ground truth to generate pairwise constraints and to evaluate the performance of dimensionality reduction techniques.

In particular, we use the 10% subset of labeled data provided by KDD-Cup, which contains 494,021 data items. This subset has 23 unique intrusion types. The distribution of the intrusion types is highly uneven, with the most common type of intrusion observed 280,790 times while the most uncommon intrusion observed two times only. Clustering imbalanced data is beyond the scope of this paper. We pick intrusion types that have more than 200 observations and for types that contain more than 300 hundred observations, we randomly sampled 300 observations for our experiment. This end up with 3195 data items from 11 distinct intrusion types. Note that the purpose of this experiment is not to propose and evaluate an accurate intrusion detector, but

Table 3
Performance on intrusion detection data ($r=10$).

	Unsupervised			20 pairs constraints				5 pairs			
Metric	*K*-means	PCA	LPP	SSK-means	SLPP	LPPSI	DSP	SSK-means	SLPP	LPPSI	DSP
F	0.6793	0.7170	0.6475	0.7156	0.6186	0.2955	**0.8183**	0.6404	0.6148	0.2838	**0.7880**
RI	0.9335	0.9449	0.9177	0.9396	0.9184	0.6778	**0.9645**	0.9251	0.9139	0.6694	**0.9575**

Table 4
F-score on half-size feature spaces.

	Unsupervised		20 pairs			5 pairs		
Dataset	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
Ionosphere	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
Sonar	0.5012	0.5423	0.5293	0.5379	**0.5873**	0.5364	0.5472	**0.5493**

to use the real-world dataset to evaluate semi-supervised dimensionality reduction techniques.

Table 3 lists the results achieved by each algorithm on the intrusion detection dataset. For 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 the dataset, where k is the number of clusters. For easy reference, we refer to them as "5(20) pairs" of constraints hereafter.

As shown in the table, SSK-means is able to improve clustering performance, compared to the baseline *K*-means, by using must-links constraints when 20 pairs of must-links are available. This means that the kernel null space projection is effective in exploring must-links. However, when only five pairs of must-links are available, the performance of SSK-means is inferior to *K*-means. This may be due to the overfitting of this kernel method given small amount of constraints. On the other hand, DSP significantly improves the clustering accuracy in all the cases, and no overfitting is observed with five pairs of constraints. These results empirically validate that DSP is able to further explore cannot-link constraints as well as the intrinsic data structure to learn a robust optimal low-dimensional embedding with even a very small amount of supervision. Furthermore, DSP notably outperforms other competitive dimensionality reduction methods for this dataset.

5.6. 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. Follow the experiment design in the last experiment, 5 and 20 pairs of constraints are alternatively generated for each cluster and each dataset. We fix the subspace dimension to be half of the original dimension. Table 4 shows the evaluation result measured in *F*-score. We observed that although *F*-score and RI have different absolute values, they show overall similar patterns for different algorithms, as evidently shown in Table 3. We thus only report results in *F*-score for clean presentation. On six out of eight datasets, DSP achieves the best *F*-scores. For the remaining two datasets, DSP still shows satisfactory *F*-scores. Most importantly, when the number of constraints is small (i.e. the five 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.

5.7. Various subspace dimensions

In this experiment, we evaluate the dimensionality reduction techniques for various subspace dimensions. For each dataset 5/10/20/30 pairs of constraints per cluster are generated following the same experiment design in previous experiments. The reduced dimensions range from 2 to 200. Figs. 4 and 5 show the results on the COIL-20 database for 3D object recognition and the multiple features dataset for handwritten digit recognition respectively. The rest datasets show similar patterns. 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.

5.8. Generalization

In this experiment, we evaluate how well DSP handles new data points on four large-scale datasets. For each dataset, we do five-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 test data points are projected to the learned subspace for further clustering evaluation. Table 5 shows the generalization performance, compared to the clustering result of test 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 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.

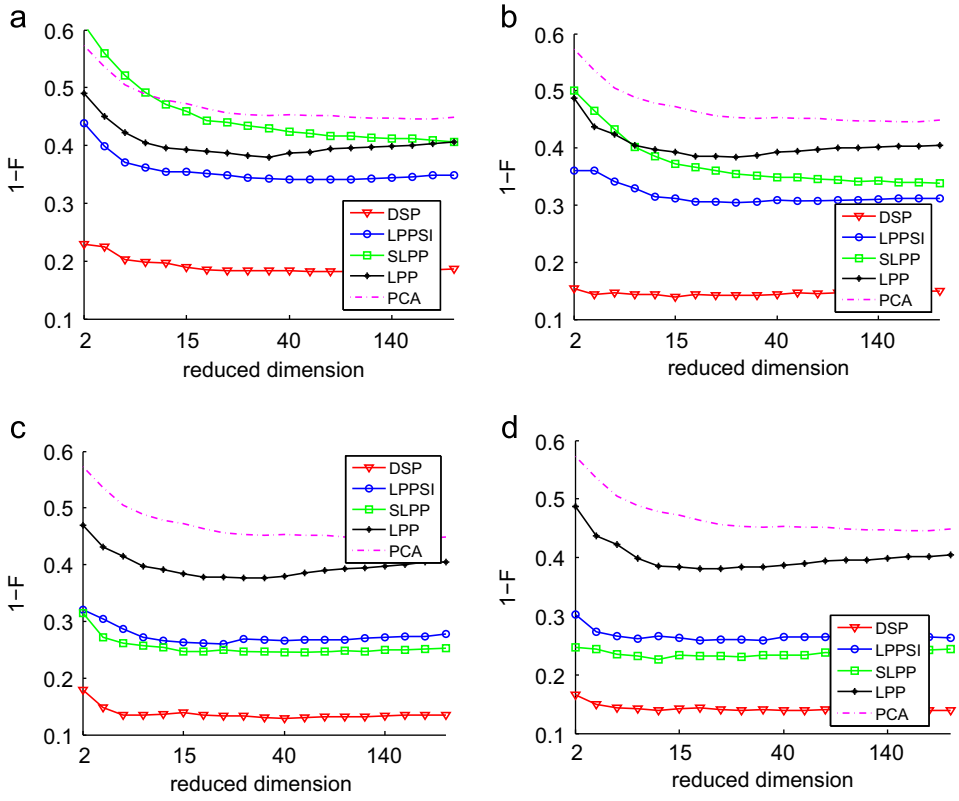

Fig. 4. Error rate vs. reduced dimensions for 3D object recognition. (a) 5 pairs, (b) 10 pairs, (c) 20 pairs, (d) 30 pairs.

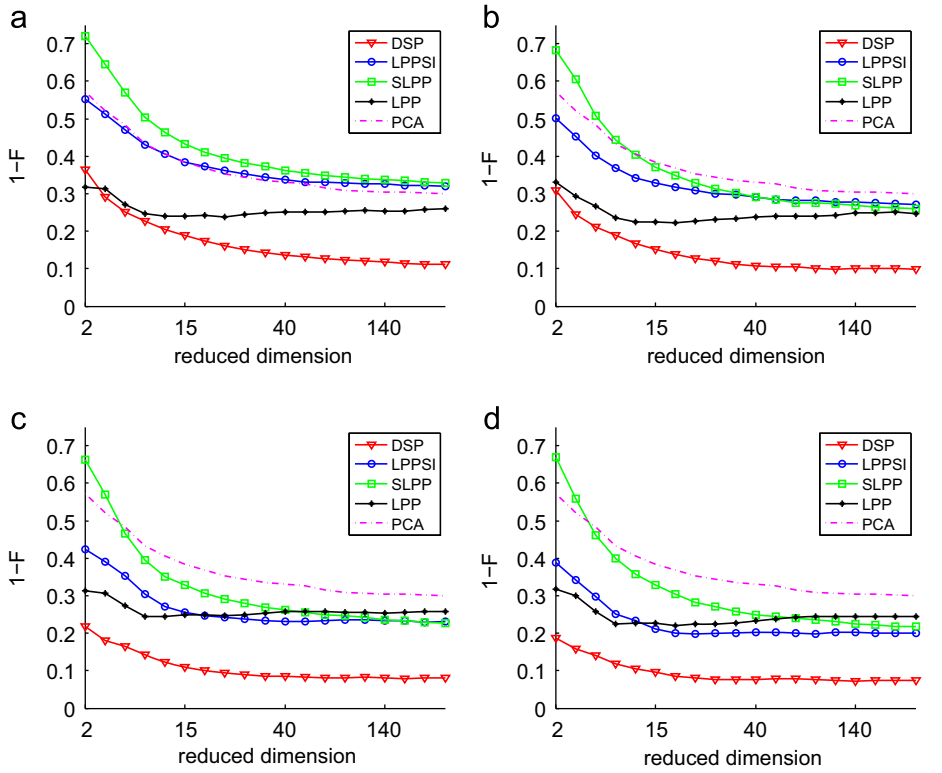

Fig. 5. Error rate vs. reduced dimensions for handwritten digit recognition. (a) 5 pairs, (b) 10 pairs, (c) 20 pairs, (d) 30 pairs.

5.9. *Running time*

We evaluate the running time of DSP and compare to competing algorithms. Because DSP learns a linearly transformation matrix, the learning can be performed offline in the training phase. In the online dimension reduction phase, the learned transformation matrix is adopted by a simple matrix multiplication operation. The supervised method SLPP works in the same manner. The competing

Table 5
F-score for generalization (*r*: subspace dimensionality).

Dataset	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)

Table 6
CPU time comparison (s).

	Offline training		Online reduction				
	Semi	Supervised	Unsupervised		Semi		Supervised
Dataset	DSP	SLPP	PCA	LPP	LPPSI	DSP	SLPP
Multiple features	26.359	9.453	11.125	11.469	14.203	0.031	0.031
COIL-20	5.031	3.266	1.516	3.531	29.406	0.000	0.000
Isolet	22.938	14.125	20.984	129.656	235.828	0.063	0.063
Pendigit	6.375	2.047	0.078	182.125	17.016	0.000	0.000

semi-supervised method LPPSI as well as the unsupervised methods perform the learning online. All the algorithms are implemented in Matlab and are evaluated on a 32bit Windows server with 3.75 GB of ram. For each dataset, one-fifth of the data are used for training and all the data are used for online reduction. Evaluation results are listed in Table 6. As the table shows, DSP and SLPP achieve the minimum online reduction time, and the saving of running time is significant. The training time for DSP is longer than the training time for SLPP. This is due to difference in the size of matrices involved in the eigenvalue problems of the two algorithms. As long as the training in performed offline, a little slowdown is training is tolerable.

6. Conclusion

We present a novel semi-supervised dimensionality reduction technique, named dual subspace projections (DSP), to cope with the learning deficiencies and computational difficulties incurred by high-dimensional data. We study two types of constraints that indicate whether or not pairs of data points originate from the same class. The 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 the intrinsic structure of data is more evident, and thus eases the subsequent data analysis. The method handles nonlinearity of data using kernel techniques, but is able to learn a linear transformation matrix. Thus, generalization to new data points is straightforward and efficient. This method is also robust to over-fitting and can benefit from constraints when only a few are available. We also present a new semi-supervised clustering technique, named subspace kernel *K*-means (SSK-means), which extends traditional kernel *K*-means by exploring must-link constraints as an intermediate step. Experiments on real datasets from multiple domains clearly demonstrate that significant improvement in learning accuracy can be achieved after our dimensionality reduction technique is employed with only a few user-supplied constraints.

Appendix A

We 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.

Proof. Let $(\phi(\mathbf{x}_i),\phi(\mathbf{x}_i'))$ be the ith 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

$$\hat{\phi}(\mathbf{x}) = \mathbf{P}\phi(\mathbf{x}) \tag{29}$$

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

$$\begin{aligned}\hat{\phi}(\mathbf{x}_i)-\hat{\phi}(\mathbf{x}_i') &= \mathbf{P}(\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i')) = (\mathbf{I}-\mathbf{U})(\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i')) \\ &= (\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i'))-\mathbf{U}(\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i')) \\ &= (\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i'))-(\phi(\mathbf{x}_i)-\phi(\mathbf{x}_i')) = 0 \qquad \square\end{aligned} \tag{30}$$

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

$$\hat{\phi}(\mathbf{x}_i) = \hat{\phi}(\mathbf{x}_i') \tag{31}$$

Thus the two points are mapped to the same point.

References

[1] S. An, W. Liu, S. Venkatesh, Exploiting side information in locality preserving projection, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[2] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning distance functions using equivalence relations, in: International Conference on Machine Learning (ICML), 2003, pp. 11–18.
[3] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, European Conference on Computer Vision (ECCV), vol. 1064, 1996, pp. 45–58.
[4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[5] H. Cevikalp, J. Verbeek, F. Jurie, A. Kläser, Semi-supervised dimensionality reduction using pairwise equivalence constraints, in: International Conference on Computer Vision Theory and Applications, 2008, pp. 489–496.
[6] R. Chatpatanasiri, B. Kijsirikul, A unified semi-supervised dimensionality reduction framework for manifold learning, Neurocomputing 73 (10–12) (2010) 1631–1640.
[7] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, 1997.
[8] J.J. Dai, L. Lieu, D. Rocke, Dimension reduction for classification with gene expression microarray data, Statistical Applications in Genetics and Molecular Biology 5 (2006).
[9] V. De Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 705–712.
[10] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (6) (1990) 391–407.
[11] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., , 1996.
[12] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2001.

[13] X. He, P. Niyogi, Locality preserving projections, in: The Neural Information Processing Systems (NIPS), 2003.
[14] S.C.H. Hoi, W. Liu, M.R. Lyu, W.-Y. Ma, Learning distance metrics with contextual constraints for image retrieval, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 2072–2078.
[15] E. Kokiopoulou, P. Frossard, Semantic coding by supervised dimensionality reduction, IEEE Transactions on Multimedia 10 (5) (2008) 806–818.
[16] D. Koller, M. Sahami, Toward optimal feature selection, in: International Conference on Machine Learning (ICML), 1996, pp. 284–292.
[17] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K.R. Mullers, Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing IX, 1999. Signal Processing Society Workshop, 1999, pp. 41–48.
[18] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine 2 (6) (1901) 559–572.
[19] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics (Texts in Applied Mathematics), Springer-Verlag New York, Inc., 2006.
[20] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326 12 2000.
[21] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, NY, USA, 2004.
[22] G.W. Stewart, Matrix Algorithms, Volume II: Eigensystems. SIAM: Society for Industrial and Applied Mathematics, first ed., August 2001.
[23] M. Sun, C. Liu, J. Yang, Z. Jin, J. Yang, A two-step framework for highly nonlinear data unfolding, Neurocomputing 73 (10–12) (2010) 1801–1807.
[24] J.A.K. Suykens, Data visualization and dimensionality reduction using kernel maps with a reference point, IEEE Transactions on Neural Networks 19 (9) (2008) 1501–1517.
[25] W. Tang, H. Xiong, S. Zhong, J. Wu, Enhancing semi-supervised clustering: a feature projection perspective, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2007, pp. 707–716.
[26] J.B. Tenenbaum, V. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[27] O. Tuzel, F. Porikli, P. Meer, Kernel methods for weakly supervised mean shift clustering, in: International Conference on Computer Vision (ICCV), 2009, pp. 59–68.
[28] M. Verleysen, Learning high-dimensional data, in: Limitations and Future Trends in Neural Computation, 2003, pp. 141–162.
[29] H. Yu, J. Yang, A direct lda algorithm for high-dimensional data with application to face recognition, Pattern Recognition 34 (2001) 2067–2070.
[30] D. Zhang, Z.-H. Zhou, S. Chen, Semi-supervised dimensionality reduction, in: SIAM Conference on Data Mining (SDM), 2007.

Su Yan is currently a Postdoctoral scholar at IBM Almaden Research Center. She obtained her PhD degree in Information Sciences and Technology from the Pennsylvania State University in 2010. Before that, she obtained an MS and a BS in Electrical Engineering from Jilin University. During her PhD study, she worked as a research intern at Mitsubishi Electric Research Laboratories (MERL) in 2009 and a summer intern at IBM Silicon Valley Lab (SVL) in 2008 and 2007. Her research interests include Machine Learning, Data Mining, Text Analysis, Information Extraction, Information Retrieval, Bibliometrics and Databases.

Sofien Bouaziz is currently a PhD student in the Computer Graphics and Geometry Laboratory at the Ecole Polytechnique Fédérale de Lausanne (EPFL) under the supervision of Prof. Mark Pauly. He received his MSc degree in Computer Science from EPFL in 2009 and completed his master thesis at the Imaging Group of Mitsubishi Electric Research Laboratories (MERL). After his graduation and before starting his PhD in 2010, he joined e-on software as an R&D software engineer. His research interests include computer graphics, computer vision and machine learning.

Dongwon Lee is currently an associate professor of College of Information Sciences and Technology (IST) in The Pennsylvania State University. He obtained a BS from Korea University in 1993, an MS from Columbia University in 1995, and a PhD from UCLA in 2002, all in Computer Science. In between MS and PhD, he has worked at AT&T Bell Labs from 1995 to 1997. His research interests include Digital Libraries, Bibliometrics, Databases, Data Mining, Web Search and Analysis, and Semantic Web Services. He has (co-)authored about 90 scholarly articles in selected conferences or journals.

Jesse Barlow is currently a professor of Computer Science and Engineering at The Pennsylvania State University where he has worked since 1981. He obtained a BS from the University of Kansas in 1977 in Mathematics and Computer Science, an MS from Northwestern University in 1980 in Statistics, and a PhD from Northwestern in Computer Science in 1981. He has been a visiting faculty at New York University and the University of Manchester. His research interests include Numerical Linear Algebra, numerical problems in signal and image processing, and numerical aspects of computer arithmetic. He has written numerous technical articles in these fields and serves of the board of the SIAM Journal on Matrix Analysis and Applications and of Computational Statistics and Data Analysis.