Interpretable Discriminative Dimensionality Reduction and Feature Selection on the Manifold

Babak Hosseini[1] ✉[0000−0003−3335−4289], Barbara Hammer[1]

CITEC centre of excellence, Bielefeld University
Bielefeld, Germany
{bhosseini,bhammer}@techfak.uni-bielefeld.de

Abstract. Dimensionality reduction (DR) on the manifold includes effective methods which project the data from an implicit relational space onto a vectorial space. Regardless of the achievements in this area, these algorithms suffer from the lack of interpretation of the projection dimensions. Therefore, it is often difficult to explain the physical meaning behind the embedding dimensions. In this research, we propose the interpretable kernel DR algorithm (I-KDR) as a new algorithm which maps the data from the feature space to a lower dimensional space where the classes are more condensed with less overlapping. Besides, the algorithm creates the dimensions upon local contributions of the data samples, which makes it easier to interpret them by class labels. Additionally, we efficiently fuse the DR with feature selection task to select the most relevant features of the original space to the discriminative objective. Based on the empirical evidence, I-KDR provides better interpretations for embedding dimensions as well as higher discriminative performance in the embedded space compared to the state-of-the-art and popular DR algorithms.

Keywords: Dimensionality reduction · Interpretability · Supervised.

1 Introduction

Dimensionality reduction (DR) is an essential preprocessing phase in the application of many algorithms in machine learning and data analytics. The general goal in any DR approach is to obtain an embedding to transfer the data from the original high-dimensional (HD) space to a low-dimension (LD) space, such that this projection preserves the vital information about the data distribution[23]. It is common to split the dimensionality reduction methods into two groups of unsupervised and supervised algorithms. The first group includes methods such as Principal Component Analysis (PCA) [13] which finds a new embedding space in which the dimensions are sorted based on the maximum data variation they can achieve, or locally linear embedding (LLE) [23] that focuses on preserving the relational structure of data points in the local neighborhoods of the space throughout an embedding.

The second group of algorithms, known as supervised (discriminative) DR methods, assume that data classes can obtain the same or even better separations

in an intrinsic LD space. As a popular supervised algorithm, Linear Discriminant Analysis (LDA) [19] tries to find a mapping which increases the distance between the class centroids while preserving the intra-class variations. Its subsequent algorithms such as LLDA [14] and CPM [29] tried to relax the constraints on within-class variations to project the sub-clusters to the LD space more efficiently.

It is possible to consider an implicit mapping of data to a high-dimensional reproducing kernel Hilbert space (RKHS) primarily to obtain a relational representation of the non-vectorial or structured data distributions. Consequently, a branch of DR algorithms (kernel-DR) is focused on kernel-based data representations to transfer the data from the original RKHS to a vectorial space. This projection can become significant especially when it makes the application of many vectorial algorithms possible on LD embedding of such data. The most famous kernel-DR algorithms are Kernelized PCA (K-PCA) and K-FDA [19] which are the kernelized versions of PCA and LDA algorithms respectively. In these methods and many other kernel-DR algorithms, it is common to construct the embedding dimensions upon different weighted combinations of data points in the original RKHS. Other notable examples of kernel-based methods include algorithms such as KDR [8], KEDR [1], and LDR [24].

Additionally, by assuming a set of non-linear mappings to different sub-spaces in the feature space, it is possible to obtain one specific kernel representation for each dimension of the data [6, 10]. Consequently, a specific group of methods tried to apply DR frameworks also to feature selection tasks on manifolds [16, 12].

One of the important practical concerns regarding dimensionality reduction is the interpretation of new dimensions. It is common to observe in many DR methods that the embedding dimensions are constructed upon arbitrary combinations of many uncorrelated physical dimensions [25, 4]. Such occasions can make the interpretation of these dimensions difficult or impossible. Such condition becomes even more severe for kernel-DR methods where the embedding dimensions are an implicit combination of data points in RKHS. For instance methods similar to K-PCA, the embedding vectors almost use weighted combination of all data points from all the classes. Hence, it would be difficult to relate any of the dimensions to any class of data (Figure 1(a)). Furthermore, a high correlation between embedding directions can be found when considering the class-contributions in them (Figure 1(b)).

As an improvement, sparse K-PCA [27] applies an l_1-norm sparsity objective to form embedding vectors from sparse combinations of training samples. However, these samples still belong to different classes which makes the resulting embeddings weak according to the class-based interpretation (Figure 1).

1.1 Motivation

As discussed in the previous paragraphs, one crucial challenge for kernel-DR algorithms is the interpretation of their projection dimensions. Based on the relation of these dimensions to the selection of data points, it is logical to focus

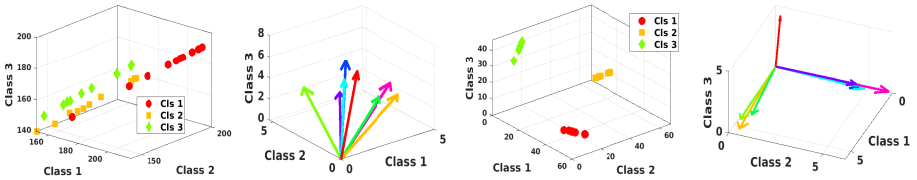

Fig. 1. When embedding vectors use all data points: (a) its projection on class-labels is coupled and (b) the embedding vectors are highly correlated in the label space. A class-based interpretable embedding: (c) provides a more distinct projection on class labels and (d) its dimensions can be distinguished and explained based on class labels.

on having each selection linked to mostly one class of data. This strategy can lead to the class-based interpretation as in Figure 1(c)(d).

Besides, current kernel-DR methods cannot efficiently embed the multi-cluster data classes to an LD space such that the clusters could still be separated from each other. In particular, they suffer from considering the local distributions inside the classes.

Based on the current state-of-the-art, the research in kernel-DR is always distinct from feature selection on the manifold. Although in some research, these concerns are employed in a single framework [16, 12], the DR aspect of the problem was not well investigated. Nevertheless, in particular for discriminative tasks, these two aspects should act as each other's complements in a single framework.

1.2 Contributions

In this work, we propose a novel discriminative dimensionality reduction method which projects the data from an implicit RKHS space to a low-dimension vectorial space. Besides, it can join this embedding with feature selection in case of having multiple representations for the data on the manifolds. We can summarize our contributions as follows:

- We introduce the class-based interpretation concern for the kernel-DR frameworks through which the embedding dimensions can be explained according to the classes they most represent.
- We show that focusing on the within-class local similarities and between-class dissimilarities can provide a more discriminative embedding.
- We fuse feature selection with our kernel-DR framework which leads to a more discriminative feature selection compared to the state-of-the-art.

In the rest of this paper, we provide preliminaries in Sec. 2 and discuss our discriminative kernel-DR framework in Sec. 3. The optimization steps and the experimental results are discussed in Sec. 3.5 and Sec. 4 respectively. We summarize our findings in the conclusion section.

2 Preliminaries

2.1 Notations

We denote the matrix of training data by $\mathbf{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \mathbb{R}^{d\times N}$, and the corresponding class label matrix is given as $\mathbf{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N] \in \{0,1\}^{c\times N}$. Each $\boldsymbol{h}_i$ is a zero vector except in its q-th entry where $h_{qi} = 1$ if $\boldsymbol{x}_i$ belongs to class q in a c-class setting. In general, for a given matrix $\mathbf{A}$, $\boldsymbol{a}_i$ denotes its i-th column, $\mathbf{A}(j,:)$ denotes its j-th row, and a_{ji} refers to the j-th entry in $\boldsymbol{a}_i$.

2.2 Kernel-based Dimensionality Reduction

Assume there exists an implicit non-linear mapping $\Phi(\mathbf{X})$ corresponding to the mapping of $\mathbf{X}$ into an RKHS, which corresponds to a kernel matrix $\mathcal{K}(\mathbf{X},\mathbf{X}) = \Phi^\top(\mathbf{X})\Phi(\mathbf{X})$. Generally, a kernel-DR algorithm tries to obtain an embedding $\boldsymbol{\gamma} = \mathbf{U}^\top\Phi(\boldsymbol{x})$ as a mapping from the features space to an LD space. Since the dimensions of $\Phi(\boldsymbol{x})$ are not directly accessible in the feature space, it is common to assume embedding dimensions are constructed as

$$\mathbf{U} = \Phi(\mathbf{X})\mathbf{A}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times k}$. Hence, the matrix $\mathbf{A}$ projects the data from the HD feature space to a k-dimensional space, where each embedding vector $\boldsymbol{a}_i$ is a combination of the training samples in RKHS.

Regarding the above, the K-PCA method preserves the variance of the reconstruction and to obtain embedding dimensions which are orthogonal and sorted based on their maximum variations. To that aim, K-PCA uses the following optimization:

$$\begin{aligned} &\min_{\mathbf{A}} \|\Phi(\mathbf{X}) - \Phi(\mathbf{X})\mathbf{A}\mathbf{A}^\top\Phi(\mathbf{X})^\top\Phi(\mathbf{X})\|_F^2 \\ &\text{s.t.} \;\; \mathbf{A}^\top\Phi(\mathbf{X})^\top\Phi(\mathbf{X})\mathbf{A} = \mathbf{I}, \end{aligned} \tag{2}$$

Although K-PCA is a powerful preprocessing algorithm to eliminate the low-variate dimensions, it does not have any direct focus on the discrimination of the embedded data classes. Also, each embedding vectors ν_i consists of both positive and negative contributions from all training samples which makes their interpretation difficult.

On the other hand, the K-FDA algorithm tries to obtain an embedding $\mathbf{W}$ which increases the between-class covariance matrix $\mathbf{S}_\mathbf{B}^\phi$ while preserving the total within-class covariance matrix $\mathbf{S}_\mathbf{W}^\phi$ in RKHS [19]. It uses the following optimization framework:

$$\max_{\mathbf{W}} \mathrm{Tr}(\mathbf{W}^\top\mathbf{S}_\mathbf{B}\mathbf{W}) \qquad \text{s.t.} \mathbf{W}^\top\mathbf{S}_\mathbf{W}\mathbf{W} = \mathbf{I}, \tag{3}$$

where $\mathbf{W}$ has a structure analogous to Eq. (1). Regardless of its supervised performance, the constraint on intra-class variances can become a critical weakness when there are sub-clusters in each data class. In such cases, the constraint in Eq. (3) cause considerable overlapping between different classes.

Our proposed framework improves the state-of-the-art in both discriminative kernel-DR and class-based interpretation of embedding dimensions.

3 Interpretable Discriminative Dimensionality Reduction

We want to obtain the embedding

$$\boldsymbol{\gamma} = \mathbf{A}^\top \Phi(\mathbf{X})^\top \Phi(\boldsymbol{x}) \qquad \boldsymbol{\Gamma} \in \mathbb{R}^k \tag{4}$$

as a projection from the original implicit RKHS to a k-dimensional explicit space which also preserves the essential characteristics of $\mathbf{X}$ in the original space.

Definition 1 *The embedding vector $\Phi(\mathbf{X})\boldsymbol{a}_i$ is class-based interpretable if we have $\frac{\mathbf{H}(q|h_{qi}=1,:)\boldsymbol{a}_i}{\|\mathbf{H}\boldsymbol{a}_i\|_1} \approx 1$, and it acts as the projection of data points on class q.*

In other words, $\Phi(\mathbf{X})\boldsymbol{a}_i$ can be interpreted as a projection to class q if it is constructed only from that class of data. Although Definition 1 considers an ideal situation regarding the interpretability of an embedding dimension, we consider the value of

$$\mathbf{H}(q|h_{qi} = 1, :)\boldsymbol{a}_i / \|\mathbf{H}\boldsymbol{a}_i\|_1 \tag{5}$$

as a measure of class-based interpretation as well. To be more specific regarding our framework, we aim for the following objectives:

O1: Increasing the class-based interpretation of embedding dimensions.

O2: The embedding should make the classes more separated in the LD space.

O3: The classes should be locally more condensed in the embedded space.

O4: The DR framework should also support the feature selection objective if a multiple kernel representation is provided.

Therefore, we formulate the following optimization scheme w.r.t. all the above objectives:

$$\begin{aligned} &\min_{\mathbf{A},\boldsymbol{\beta}} \ \mathcal{J}_{Sim} + \lambda \mathcal{J}_{Dis} + \mu \mathcal{J}_{Ip} \\ &\text{s.t.} \ \sum_{m=1}^{d} \beta_m = 1, \ \sum_{j=1}^{N} a_{ji} = 1, \forall i \\ &\quad a_{ij}, \beta_i \in \mathbb{R}^+, \ \ \forall ij. \end{aligned} \tag{6}$$

In Eq. (6), the cost functions $\mathcal{J}_{Dis}$, $\mathcal{J}_{Ip}$, and $\mathcal{J}_{Sim}$ and the constraints on the optimization variables are designed to fulfill our research objectives **O1-O4**. In the following sub-sections, we explain each specific term in our framework in detail and provide the rationales behind their definitions.

3.1 Interpretability of the Dimensions

In Eq. (4), each dimension $\boldsymbol{a}_i$ of the embedding is composed of a weighted selection of data points in RKHS. In K-PCA, typically all $a_{ji}, \forall j = 1, \ldots, N$ have non-zero values. More specifically, for each $\boldsymbol{a}_i$, a wide range of training data from different classes are selected with large weights which weaken the interpretation of $\boldsymbol{a}_i$ regarding the class to which it could be related.

To make each $\boldsymbol{a}_i$ more interpretable in our framework, we propose the cost function $\mathcal{J}_{Ip}$ that its minimization enforces $\boldsymbol{a}_i$ to be constructed using similar

samples in the RKHS:

$$\mathcal{J}_{Ip}(\mathbf{X},\mathbf{A}) = \tfrac{1}{2}\sum_{i=1}^{k}\sum_{s,t=1}^{N} a_{si}a_{ti}\|\Phi(\boldsymbol{x}_s)-\Phi(\boldsymbol{x}_t)\|_2^2, \tag{7}$$

where we restrict $a_{ij} \geq 0, \forall ij$. We call $\mathcal{J}_{Ip}$ as the interpretability term (Ip-term) which is an unsupervised function and independent from the value of $\mathbf{H}$. The Ip-term enforces each embedding dimension $\boldsymbol{a}_i$ to use samples in $\Phi(\mathbf{X})$ that are located in a local neighborhood of each other in RKHS (Figure 2) by introducing a penalty term $a_{si}a_{ti}\|\Phi(\boldsymbol{x}_s)-\Phi(\boldsymbol{x}_t)\|_2^2$ on its entries. Resulting from this term along with the non-negativity constraint on $\mathbf{A}$, non-zero entries of $\boldsymbol{a}_i$ correspond to the neighboring points such as (s,t) where their pairwise distance $\|\Phi(\boldsymbol{x}_s)-\Phi(\boldsymbol{x}_t)\|_2^2$ is small. Furthermore, although Ip-term does not employ the label information, by assuming a smooth labeling for the data, this regularization term constructs each $\boldsymbol{a}_i$ by contributions from more likely one particular class. Therefore, as a solution to our first research objective (**O1**), using Ip-term improves the class-based interpretation of $\boldsymbol{a}_i$ to relate it a sub-group of data points mostly belonging to one specific class of data (Eq. (5)).

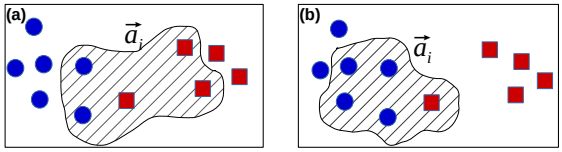

Fig. 2. Effect of using $\mathcal{J}_{Ip}$ on the formation of an embedding vector $\boldsymbol{a}_i$ as the weighted combination of selected data points (inside the hatched area) in the RKHS. (a): Without using $\mathcal{J}_{Ip}$, the learned $\boldsymbol{a}_i$ cannot be assigned to either of $\{circle, square\}$ classes. (b): After employing $\mathcal{J}_{Ip}$, the formed $\boldsymbol{a}_i$ can almost be interpreted by the *circle* class.

3.2 Inter-class dissimilarity

Regarding our second objective (**O2**), we focus on increasing the inter-class dissimilarities in the LD space which makes the embedded classes more distinct. To that aim, we define the loss term $\mathcal{J}_{Dis}$ as

$$\begin{aligned}&\mathcal{J}_{Dis}(\mathbf{X},\mathbf{H},\mathbf{A}) = \\ &\mathrm{Tr}(\overline{\mathbf{H}}^{\top}\mathbf{H}\Phi(\mathbf{X})^{\top}\Phi(\mathbf{X})\mathbf{A}\mathbf{A}^{\top}\Phi(\mathbf{X})^{\top}\Phi(\mathbf{X})),\end{aligned} \tag{8}$$

where $\overline{\mathbf{H}}$ is the logical complement of $\mathbf{H}$. Throughout simple algebraic operations, we can show that Eq. (8) is the reformulation of

$$\sum_{i}\sum_{j|\boldsymbol{h}_j\neq\boldsymbol{h}_i}\langle\mathbf{A}^{\top}\Phi(\mathbf{X})^{\top}\Phi(\boldsymbol{x}_i),\mathbf{A}^{\top}\Phi(\mathbf{X})^{\top}\Phi(\boldsymbol{x}_j)\rangle. \tag{9}$$

Hence, minimizing $\mathcal{J}_{Dis}$ motivates the global separation of the classes in the embedded space by reducing the similarity between their projected vectors $\mathbf{A}^\top \Phi(\mathbf{X})^\top \Phi(\boldsymbol{x})$.

3.3 Intra-class similarities

Even though the introduced cost term J_{Dis} helps the embedded classes to obtain more distance from each other, it still does not consider the intra-class similarities which concerns our third objective (**O3**). It is important to note that we want to make the projected vectors $\boldsymbol{\gamma}_i$ of each class more similar to each other, while still preserving the local structure of the class respecting the possible sub-classes. This characteristic works against the drawback of K-FDA when facing distinct sub-classes as pointed out by [17].

To address the above concern, we proposed the following cost function

$$\mathcal{J}_{Sim} = \sum_{i=1}^{N} (\mathbf{H}(q|h_{qi} = 1, :)\mathbf{A}\mathbf{A}^\top \Phi(\mathbf{X})^\top \Phi(\boldsymbol{x}_i) - 1)^2, \tag{10}$$

in which q is the class to which $\boldsymbol{x}_i$ belongs. Furthermore, based on Eq. (6), we apply an affine constraint on columns of $\mathbf{A}$ as $\|\boldsymbol{a}_s\|_1 = 1, \forall s = 1, \ldots, N$. By combining Eq. (10) with $\boldsymbol{\gamma}_i$ from Eq. (4) we have

$$\mathcal{J}_{Sim} = \sum_{i=1}^{N} (\mathbf{H}(q|h_{qi} = 1, :)\mathbf{A}\boldsymbol{\gamma}_i - 1)^2, \tag{11}$$

which applies constraints on columns of $\mathbf{A}$ corresponding to large entries of $\boldsymbol{\gamma}_i$. Specifically, those constraints aim the entries which are related to the data points which have the same label as $\boldsymbol{x}_i$. For instance, if γ_{si} has a relatively large value, minimizing $\mathcal{J}_{Sim}$ optimizes the entries a_{js} where $\boldsymbol{h}_j = \boldsymbol{h}_i$. Besides, the applied l_1-norm sparsity constraint $\|\boldsymbol{a}_s\|_1 = 1$ enforces some entries in $\boldsymbol{a}_s$ to shrink near to zero. Therefore, it is simple to conclude that these entries would mostly include a_{js} where $\boldsymbol{h}_j \neq \boldsymbol{h}_i$.

On the other hand, $\gamma_{si} = \sum_{t=1}^{N} a_{ts} \Phi(\boldsymbol{x}_t)^\top \Phi(\boldsymbol{x}_i)$. Hence, Having the l_1-norm of $\boldsymbol{a}_s$ restricted along with its non-negativity constraint naturally motivates the optimization process to assign large values to entries a_{ts} corresponding to data points $\boldsymbol{x}_t$ with large $\Phi(\boldsymbol{x}_t)^\top \Phi(\boldsymbol{x}_i)$. In other words, $\boldsymbol{a}_s$ selects the nearby data points of $\boldsymbol{x}_i$ as its most similar neighbors. Combining this finding with our first conclusion about the effect of Eq. (10), along with the localization role of $\mathcal{J}_{Ip}$, minimizing $\mathcal{J}_{Sim}$ helps each data point $\boldsymbol{x}_i$ to be encoded in particular by its nearby embedding vectors $\boldsymbol{a}_s$, which are also constructed mostly by the same-class of samples in the vicinity of $\boldsymbol{x}_i$ (**O1**). Consequently, the data points from each local sub-class are embedded by similar sets of columns in $\mathbf{A}$ and are mapped into a local neighborhood in the LD space. In other words, This embedding increases the intra-class similarities for the projected columns in $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N]$.

3.4 Feature Selection on the Manifold

It is a feasible assumption for any structured and non-structure $\mathbf{X}$ to have d different kernel representations available [2], such that each $\mathcal{K}_m(\mathbf{X},\mathbf{X}), \forall m = 1,\dots,d$, maps the m-th dimension of the original data into an RKHS or is derived from the m-th descriptor (e.g., for images). Given the above, we can assume

$$\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1^\top(\boldsymbol{x}),\dots,\phi_d^\top(\boldsymbol{x})]^\top, \tag{12}$$

where each $\phi_m : \mathbb{R} \to \mathbb{R}^{f_m}, \forall m = 1,\dots,d$ represents an implicit mapping from the original space to a subspace of the RKHS, such that $\mathcal{K}_m(\boldsymbol{x}_t,\boldsymbol{x}_s) = \phi_m^\top(\boldsymbol{x}_t)\phi_m(\boldsymbol{x}_s)$. Therefore, we can consider a diagonal matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$ which provides scaling of the RKHS by

$$\hat{\boldsymbol{\Phi}}(\boldsymbol{x}) = \mathbf{B}\boldsymbol{\Phi}(\mathbf{X}) = [\sqrt{\beta_1}\phi_1^\top(\boldsymbol{x}),\cdots,\sqrt{\beta_d}\phi_d^\top(\boldsymbol{x})]^\top, \tag{13}$$

where $\boldsymbol{\beta}$ is the vector of combination weights derived from diagonal entries of $\mathbf{B}$. We can compute the weighted kernel matrix $\hat{\mathcal{K}}$ corresponding to $\hat{\boldsymbol{\Phi}}(\mathbf{X})$ as

$$\hat{\mathcal{K}}(\boldsymbol{x}_t,\boldsymbol{x}_s) = \textstyle\sum_{m=1}^{d} \beta_m \mathcal{K}_m(\boldsymbol{x}_t,\boldsymbol{x}_s). \tag{14}$$

Additionally, we apply a non-negativity constraint on entries of $\boldsymbol{\beta}$ as $\beta_i \geq 0$ to make the resulted kernel weights interpreted as the relative importance of each kernel in the weighted representation $\hat{\boldsymbol{\Phi}}(\mathbf{X})$ [10]. Consequently, we can obtain a feature selection profile by sorting entries of $\boldsymbol{\beta}$ based on their magnitude. For the ease of reading, in the rest of the paper, we denote $\hat{\mathcal{K}}(\mathbf{X},\mathbf{X})$ and $\mathcal{K}_i(\mathbf{X},\mathbf{X})$ by $\hat{\mathcal{K}}$ and $\mathcal{K}_i$ respectively.

Substituting $\boldsymbol{\Phi}(\mathbf{X})$ by $\hat{\boldsymbol{\Phi}}(\mathbf{X})$ in the definitions of $\mathcal{J}_{Dis}$, $\mathcal{J}_{Ip}$, and $\mathcal{J}_{Sim}$ reformulates them also as a function of $\mathbf{B}$. Therefore, minimizing those terms also optimizes the value of $\mathbf{B}$ regarding their specific purposes. Furthermore, we apply an l_1-norm restriction on the value of $\mathbf{B}$ as the affine constraint $\sum_{m=1}^{d}\beta_m = 1$. This constraint prevents $\boldsymbol{\beta}$ from becoming a vector of zeros as the trivial solution and additionally results in a sparse feature selection to reduce the redundancies between different kernel representations [22]. We can claim that by using $\hat{\boldsymbol{\Phi}}(\mathbf{X})$ in each of the defined terms, the resulted feature selection also complies with those specific characteristics. In the next section, we discuss the optimization scheme of Eq. (6).

3.5 Optimization Scheme

The cost function $\mathcal{J}_{Sim}$ is non-convex which makes the objective function of Eq. (6) non-convex as well. Hence, we define a variable matrix $\mathbf{S}$ and relax Eq. (6) to the following optimization problem

$$\begin{aligned}
\min_{\mathbf{A},\boldsymbol{\beta},\mathbf{S},\boldsymbol{\Gamma}}\ & \textstyle\sum_{i=1}^{N}(\mathbf{H}(q|h_{qi}=1,:)\boldsymbol{s}_i - 1)^2 \\
& +\lambda \mathrm{Tr}(\mathbf{A}^\top\hat{\mathcal{K}}\overline{\mathbf{H}}^\top\mathbf{H}\hat{\mathcal{K}}\mathbf{A}) + \mu\mathrm{Tr}(\mathbf{A}^\top\tilde{\mathcal{K}}\mathbf{A}) \\
& +\tau\|\mathbf{S}-\mathbf{A}\boldsymbol{\Gamma}\|_F^2 + \zeta\|\boldsymbol{\Gamma}-\mathbf{A}^\top\mathcal{K}\|_F^2 \\
\text{s.t.}\quad & \sum_{m=1}^{d}\beta_m = 1,\ \sum_{j=1}^{N}a_{ji} = 1, \forall i \\
& a_{ij},\beta_i \in \mathbb{R}^+,\ \ \forall ij,
\end{aligned} \tag{15}$$

in which $\tilde{\mathcal{K}} = diag(\hat{\mathcal{K}}\mathbf{1}) - \hat{\mathcal{K}}$, and the operator $diag(.)$ creates a diagonal matrix from its vector argument. The constants λ, μ are the control parameters for the role of introduced loss terms in the optimization scheme, and the constants τ, ζ should be large enough to make sure the slack variables $\mathbf{S}, \mathbf{\Gamma}$ have appropriate values. The second and third parts of the objective in Eq. (15) are reformulations of $\mathcal{J}_{Dis}$ and $\mathcal{J}_{Ip}$, which can be obtained by using the *kernel trick* and the Laplacian matrix [26]. We initialize the embedding matrix $\mathbf{A}$ using random entries and adjust its columns to have unit l_1-norm. Then, we optimize $\mathbf{\Gamma}, \mathbf{S}, \mathbf{A}$, and $\boldsymbol{\beta}$ alternatively based on the following steps.

(1) Fix $\mathbf{S}, \mathbf{A}$, and $\boldsymbol{\beta}$ and update $\mathbf{\Gamma}$ as:

$$\mathbf{\Gamma}^* = \mathbf{A}^\top \hat{\mathcal{K}}. \tag{16}$$

(2) Fix $\mathbf{\Gamma}, \mathbf{A}$, and $\boldsymbol{\beta}$ and update $\mathbf{S}$:

$$\boldsymbol{s_i}^* = \arg\min_{\boldsymbol{s_i}} \ \boldsymbol{s_i}^\top (\boldsymbol{u_i}^\top \boldsymbol{u_i} + \mathbf{I}) \boldsymbol{s_i} - 2(\boldsymbol{u_i} + \boldsymbol{\gamma}_i^\top \mathbf{A}^\top) \boldsymbol{s_i}, \tag{17}$$

where $\boldsymbol{u_i} = \mathbf{H}(q|h_{qi} = 1, :)$. This unconstrained quadratic programming has the closed-form solution

$$\boldsymbol{s_i}^* = (\boldsymbol{u_i}^\top \boldsymbol{u_i} + \mathbf{I})^{-1} (\boldsymbol{u_i} + \boldsymbol{\gamma}_i^\top \mathbf{A}^\top)^\top. \tag{18}$$

(3) Fix $\mathbf{\Gamma}, \mathbf{S}$, and $\boldsymbol{\beta}$ and update $\mathbf{A}$ as:

$$\begin{aligned} \mathbf{A}^* = \arg\min_{\mathbf{A}} \ & \lambda \mathrm{Tr}(\mathbf{A}^\top \hat{\mathcal{K}} \overline{\mathbf{H}}^\top \mathbf{H} \hat{\mathcal{K}} \mathbf{A}) + \mu \mathrm{Tr}(\mathbf{A}^\top \tilde{\mathcal{K}} \mathbf{A}) \\ & + \tau \|\mathbf{S} - \mathbf{A}\mathbf{\Gamma}\|_F^2 + \zeta \|\mathbf{\Gamma} - \mathbf{A}^\top \mathcal{K}\|_F^2 \\ \text{s.t.} \quad & \mathbf{A}^\top \mathbf{1} = \mathbf{1}, \ a_{ij} \in \mathbb{R}^+, \forall ij. \end{aligned} \tag{19}$$

Calling the objective of Eq. (19) $\mathcal{J}_{\mathbf{A}}$, it is possible to show that $\mathcal{J}_{\mathbf{A}}$ consists of convex parts and its gradient w.r.t. $\mathbf{A}$ can be computed as:

$$\nabla_{\mathbf{A}} \mathcal{J}_{\mathbf{A}} = \Omega \mathbf{A} + \Psi, \tag{20}$$

where (Ω, Ψ) can be obtained by simple algebraic operations. Therefore, we use the direction method of multipliers (ADMM) [3] by defining the Lagrangian formulation for Eq. (19):

$$\begin{aligned} & \mathcal{L}_\rho(\mathbf{A}, \mathbf{A}_+, \Delta, \delta) \\ & = \mathcal{J}_A + \tfrac{\rho}{2} \|\mathbf{A} - \mathbf{A}_+\|_2^2 + \tfrac{\rho}{2} \|\mathbf{A}^\top \mathbf{1} - \mathbf{1}\|_2^2 \\ & + tr(\Delta^\top (\mathbf{A} - \mathbf{A}_+)) + \delta^\top (\mathbf{A}^\top \mathbf{1} - \mathbf{1}), \end{aligned} \tag{21}$$

and following these steps:

$$\begin{cases} \mathbf{A}^{(t+1)} = \arg\min_{\mathbf{A}} \mathcal{L}_\rho(\mathbf{A}, \mathbf{A}_+, \Delta, \delta), \\ \mathbf{A}_+^{(t+1)} = \max(\mathbf{A}^{(t+1)} + \frac{1}{\rho} \Delta^{(t)}, 0), \\ \Delta^{(t+1)} = \Delta^{(t)} + \rho(\mathbf{A}^{(t+1)} \mathbf{1} - \mathbf{1}), \\ \delta^{(t+1)} = \delta^{(t)} + \rho(\mathbf{A}^{(t+1)} - \mathbf{A}_+^{(t+1)}), \end{cases} \tag{22}$$

In Eq. (22), $\mathbf{A}_+$ is an axillary matrix related to the non-negativity constraint, $\Delta \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\delta} \in \mathbb{R}^N$ are the Lagrangian multipliers, and $\rho \in \mathbb{R}^+$ is the penalty parameter. We update the matrix $\mathbf{A}^{(t+1)}$ based on its closed-form solution derived from having $\nabla_{\mathbf{A}}\mathcal{L}_\rho = 0$.

(4) Fix $\boldsymbol{\Gamma}, \mathbf{S}$ and $\mathbf{A}$ and update $\boldsymbol{\beta}$: By combining Eq. (14) and Eq. (15) and removing the constant terms, $\boldsymbol{\beta}$ can be updated by the following quadratic programming (QP)

$$\begin{aligned} \boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \ & \tfrac{1}{2}\boldsymbol{\beta}^\top \mathbf{Q}\boldsymbol{\beta} + \boldsymbol{v}^\top \boldsymbol{\beta}, \\ \text{s.t.} \quad & \boldsymbol{\beta}^\top \mathbf{1} = 1,\ \beta_i \in \mathbb{R}^+, \forall i. \end{aligned} \tag{23}$$

In this formulation, $\forall ij = 1, \ldots, d$:

$$\mathbf{Q}_{ij} = \lambda \mathrm{Tr}(\mathbf{A}^\top \hat{\mathcal{K}}_i \overline{\mathbf{H}}^\top \mathbf{H} \hat{\mathcal{K}}_j \mathbf{A}) + \zeta \mathrm{Tr}(\hat{\mathcal{K}}_i \mathbf{A}^\top \mathbf{A} \hat{\mathcal{K}}_j), \tag{24}$$

and

$$v_i = \mu \mathrm{Tr}(\mathbf{A}^\top \tilde{\mathcal{K}}_i \mathbf{A}) - 2\mathrm{Tr}(\boldsymbol{\Gamma}^\top \mathbf{A}^\top \hat{\mathcal{K}}_i). \tag{25}$$

The optimization problem of Eq. (23) is an instance of constraint quadratic programming and can be efficiently solved by QP solvers such as CGAL[9] or MOSEK [20].

As a result, in each iteration of the main optimization loop, we compute the closed-form solution of $\boldsymbol{\Gamma}, \mathbf{S}$ and update $\mathbf{A}, \boldsymbol{\beta}$ rapidly using the ADMM and QP solvers respectively. The precise implementation of our kernel-DR framework is available on the online repository[1]

3.6 Time Complexity of the Algorithm

In the training phase, we update $\mathbf{A}, \mathbf{S}, \boldsymbol{\Gamma}$, and $\boldsymbol{\beta}$ alternatively. For each iteration of the algorithm, the variables $\{\mathbf{A}, \mathbf{S}, \boldsymbol{\Gamma}, \boldsymbol{\beta}\}$ are updated with the time complexities of $\mathcal{O}(\mathbf{M}(k^3 + k^2N + kN^2))$, $\mathcal{O}(N(N^3 + N))$, $\mathcal{O}(kN)$, and $\mathcal{O}(d^2(kc + kN + k^2) + d(k^2 + kN) + d^2L)$ respectively, where $\mathbf{M}$ is the number of iterations which takes for the ADMM algorithm to update $\mathbf{A}$, and $\mathcal{O}(d^2L)$ is the time complexity of the QP for updating $\boldsymbol{\beta}$. In practice, values of k, c, and d are much smaller than N. Hence, the computationally expensive part of the algorithm is due to computing the inverse of $(\boldsymbol{u_i}^\top \boldsymbol{u_i} + \mathbf{I})^{-1}$ to update each column of $\mathbf{S}$. However, this particular computation is independent of update rules in the iterations, and we conduct it only once in the initialization phase of the algorithm, which considerably accelerates the convergence speed.

4 Experiments

In this section, we implement our proposed I-KDR algorithm on real-world datasets to analyze its DR and feature selection performance. For all the datasets we compute the kernels based on the Gaussian kernel function

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/\delta), \tag{26}$$

[1] https://github.com/bab-git/

Table 1. Selected datasets. {**Dim**: #dimensions, **Cls**: #classes, **Num**: #data samples}.

Dataset	Num	Dim	Cls	Dataset	Num	Dim	Cls
Yale	165	1024	15	Gli85	85	22283	2
Sonar	208	60	2	CNS	60	7129	2
Colon	62	2000	2	Dbwork	64	4702	2
20NG	4852	28299	4	XM2VTS50	1180	1024	20

in which δ denotes the average of $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ for all training samples.

4.1 Datasets

We implement our DR algorithm on real-world benchmark datasets including Yale face recognition[2], {Sonar, Dbworld} from the UCI repository[3], XM2VTS50 image dataset [18], the text datasets 20newsgroups7[4], and {Colon, Gli85, Central-Nervous-System (CNS)} from the feature selection repository[5]. For the 20newsgroups7 dataset, we choose the large topic *comp*, and for Colon and Gli35 datasets we use the first two classes. The characteristics of the datasets are reported in Table 1

We evaluate the performance of the algorithms based on the average classification accuracy with 10-fold cross-validation (CV), and we use the 1-nearest neighbor method (1-NN) to predict the label of test data based on $\boldsymbol{\Gamma}$ of the training set. Moreover, the parameters λ and μ are tuned based on conducting CV on the training sets. The same policy is applied to the selected baseline algorithms.

4.2 Dimensionality Reduction

In this section, we only evaluate the dimensionality reduction performance of our I-KDR in a single-kernel scenario, meaning that we use $\mathcal{K}$ in Eq. (15) instead of $\hat{\mathcal{K}}$, and $\boldsymbol{\beta}$ is not involved in the framework. As baseline kernel-DR methods, we choose the supervised algorithm K-FDA, LDR [24], SDR [21], KDR [7], and unsupervised DR algorithms JSE [15], SKPCA [5], and KEDR [1]. The classification results are reported in Table 2.

We can observe that I-KDR obtains better performance than baselines on almost all selected datasets. For the Colon dataset, I-KDR obtained 8.26% higher accuracy than the best approach. We can conclude that our designed elements of Eq. (6) results in better discriminative projections than other baselines. Regarding other algorithm, the supervised methods (e.g., LDR and SDR) generally outperform the unsupervised ones which is due to their advantage of using the supervised information in the trainings. For Sonar and Dbwork datasets, LDR almost achieved a performance comparative to I-KDR.

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[3] http://archive.ics.uci.edu/ml/datasets.html
[4] http://qwone.com/ jason/20Newsgroups/
[5] http://featureselection.asu.edu/datasets.php

Table 2. Classification accuracies (%) on the selected datasets.

Dataset	I-KDR	LDR	SDR	KDR	K-FDA	JSE	KEDR	SKPCA
Yale	**79.43**	72.80	71.13	69.50	67.88	66.23	64.61	60.75
Sonar	87.01	86.79	84.59	85.92	83.45	81.11	82.44	71.26
Colon	**83.37**	75.09	74.03	73.19	72.05	70.81	70.00	68.12
20NG	**85.74**	80.76	79.62	80.18	78.99	77.82	76.82	72.73
Gli85	**76.45**	72.15	70.66	69.26	67.50	65.79	66.68	61.38
CNS	**72.96**	68.77	67.09	65.84	64.61	63.21	63.96	58.93
Dbwork	88.24	87.67	86.28	84.90	83.27	81.74	80.40	77.32
XM2VTS50	**95.51**	92.67	91.62	92.17	90.88	89.52	88.55	84.86

The best result (**bold**) is according to a two-valued t-test at a 5% significance level.

In Figure 3, we compare the classification accuracy of the baselines for different numbers of selected dimensions. Based on the accuracy curves, I-KDR shows a distinct performance compared to other methods for the datasets Yale, Colon, and Gli85. Especially for the high-dimensional datasets Colon and Gli85, our DR algorithm achieves the peak of its performance for a smaller number of selected dimensions in comparison. For Sonar and Dbwork, I-KDR algorithm shows a competitive performance to the best baseline (LDR algorithm). Considering the classification accuracies for Yale dataset in Figure 3, I-KDR's curve reaches the peak accuracy of each baseline while selecting fewer dimensions for the embeddings. Regarding the baseline DR algorithms, the supervised methods generally outperform the unsupervised algorithms in both the accuracy and number of selected dimension. This finding also complies with the reported information in

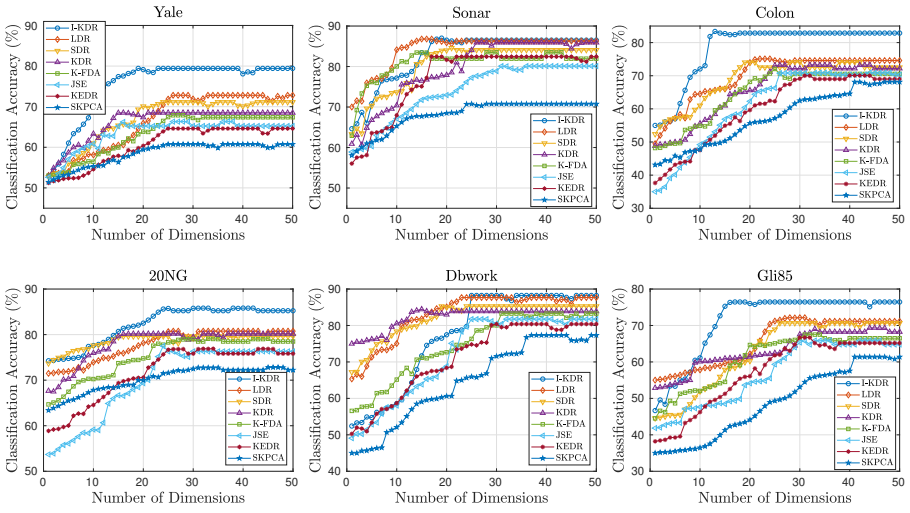

Fig. 3. Classification accuracy (%) of the baselines respect to the number of selected dimensions for the datasets Yale, Sonar, Colon, 20NG, Dbwork, and Gli85.

Table 2. Therefore, applying constraints regarding the interpretability of the DR model in I-KDR does not sacrifice its discriminative performance.

4.3 Interpretation of the Embedding Dimension

To evaluate the effect of $\mathcal{J}_{Ip}$ in Eq. (6), we use the Ip measure defined as $Ip = \frac{1}{k}\sum_{i=1}^{k}(\max_q \ \mathbf{H}(q,:)\boldsymbol{a}_i)/\|\mathbf{H}\boldsymbol{a}_i\|_1$. The Ip value considers the interpretability of each $\boldsymbol{a}_i$ based on the data points from which it is constructed. Assuming there exists considerable similarities between the class members in RKHS, a highly interpretable embedding dimension would be formed by contributions taken from mostly one class of data. In such a case, the value of Ip should grow towards 1. Table 3 reports the value of this measure for those experiments in Table 2 where computing Ip is possible. Based on the results, I-KDR obtained the most interpretable embeddings among other baselines, K-FDA has the weakest Ip performance while SKPCA and KDR are jointly the runner up methods in this Table. Regardless of the interpretation-effective sparsity term of SKPCA, its unsupervised model allows cross-class contributions to happen in the formation of the columns of $\mathbf{A}$. From another point of view, for Yale and CNS datasets, I-KDR has smaller Ip values compared to XM2VTS and 20NG datasets for instance. This difference happened due to substantial overlapping of the classes in the first group of datasets.

Additionally, to visualize the interpretation of the embeddings, we project the embedding dimensions on the label-space by computing $\mathbf{D} = \mathbf{HA} \in \mathbb{R}^{c\times k}$. Each column of $\mathbf{D}$ is a c-dimensional vector that its q-th entry explains how strong is the relation of this dimension to the class q. Figure 4 visualizes the columns of $\mathbf{D}$ for I-KDR, K-FDA, SKPCA, and KDR according to their implementations on the Sonar dataset. Each embedding was done for 10 target dimensions. Based on the results, I-KDR's embedding dimensions are almost separated into two distinct groups each of which mostly related to one class in the data. Although for SKPCA and KDR the vectors almost belong to two separate groups, they cannot be assigned to any of the classes confidently. For K-FDA, almost none of the above can be observed.

Table 3. Comparison of the Ip measure between the baselines.

Dataset	I-KDR	SKPCA	KDR	SDR	K-FDA
Yale	**0.80**	0.64	0.61	0.58	0.55
Sonar	**0.88**	0.64	0.66	0.63	0.57
Colon	**0.91**	0.72	0.69	0.66	0.63
20NG	**0.94**	0.75	0.77	0.73	0.64
Gli85	**0.84**	0.69	0.64	0.59	0.57
CNS	**0.83**	0.66	0.67	0.66	0.63
Dbwork	**0.86**	0.73	0.77	0.70	0.61
XM2VTS50	**0.96**	0.82	0.86	0.79	0.60

The best result (**bold**) is according to a two-valued t-test at a 5% significance level.

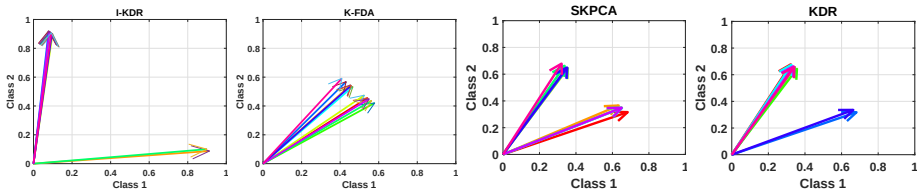

Fig. 4. Projecting the embedding dimensions on the label-space for the Sonar dataset.

4.4 Feature Selection

In order to evaluate the feature selection performance of our I-KDR algorithm, we compute Eq. (26) for each dimension of the data individually which results in a set of kernels $\{\mathcal{K}_i\}_{i=1}^{d}$ for each dataset. We feed these kernels to the optimization framework of Eq. (15) to optimize their corresponding weights in $\boldsymbol{\beta}$. Besides the classification accuracy, we also measure $\|\boldsymbol{\beta}\|_0$ to evaluate the feature selection performance of the algorithms. Accordingly, we choose the following relevant set of baselines: MKL-TR [12], MKL-DR [16], KNMF-MKL [11], and DMKL [28]. Based on Table 4, by optimizing the value of $\boldsymbol{\beta}$ in Eq. (6), I-KDR achieves better discriminations in the embedded space. Consequently, as a general trend among the datasets, I-KDR's accuracies are improved after we optimized it in the multiple kernel framework (Compared to Table 2). Regarding the number of selected features, I-KDR, MKL-TR, and DMKL obtained similar results. Even more, for some of the datasets, the baselines obtained sparser feature selections than I-KDR. Nevertheless, I-KDR demonstrates that its number of selected features are more efficient than others due to its supremacy in classification accuracies. Therefore, we can claim that I-KDA performed more efficient than others in discriminative feature selection scenarios. For CNS and Sonar dataset, I-KDR obtains competitive accuracy and feature selection performance compared to MKL-TR and DMKL, while for the Colon dataset, it outperforms the next best method (MKL-TR) with 7.73% accuracy margin. As an explanation regarding

Table 4. Comparison of classification accuracies (%) and $\|\boldsymbol{\beta}\|_0$ (in parenthesis).

Dataset	I-KDR	DMKL	MKL-TR	MKL-DR	KNMF-MKL
Yale	**83.22 (20)**	78.25 (39)	79.88 (34)	70.34 (93)	68.43 (543)
Sonar	87.91 (37)	87.53 **(34)**	87.94 (41)	70.34 (93)	68.43 (543)
Colon	**89.29** (25)	80.32 **(21)**	81.56 (34)	80.67 (67)	78.43 (1321)
20NG	**88.41** (73)	85.01 (57)	84.42 **(55)**	86.24 (384)	83.11 (14483)
Gli85	**79.65 (33)**	73.13 (54)	74.46 (50)	72.83 (79)	71.78 (10764)
CNS	76.53 (47)	76.37 **(32)**	75.84 (25)	74.23 (109)	72.43 (4872)
Dbwork	**91.98 (29)**	87.23 (41)	86.53 (46)	85.14 (85)	85.34 (1049)
XM2VTS50	**97.74 (17)**	92.76 (31)	93.84 (29)	92.88 (55)	90.89 (389)

The best result (**bold**) is according to a two-valued t-test at a 5% significance level.

the relatively high values of $\|\boldsymbol{\beta}\|_0$ for KNMF-MKL, this algorithm uses a DR model, but it does not have a discriminative objective in its optimization.

5 Conclusion

In this paper, we proposed a novel algorithm to perform discriminative dimensionality reduction on the manifold. Our I-KDR method constructs its embedding dimensions by selecting data points from local neighborhoods in the RKHS. This strategy results in embeddings with better class-based interpretations for their bases. Besides, by focusing on within-class local similarities and between-class dissimilarities, our method improves the separation of the classes in the projected space. The I-KDR algorithm has a bi-convex optimization problem, and we use the alternating optimization framework to solve it efficiently. Furthermore, our approach can fuse the feature selection and dimensionality reduction for RKHS. Our empirical results show that I-KDR outperforms other relevant baselines in both DR and feature selection scenarios.

References

1. Álvarez-Meza, A.M., Lee, J.A., Verleysen, M., Castellanos-Dominguez, G.: Kernel-based dimensionality reduction using renyi's α-entropy measures of similarity. Neurocomputing **222**, 36–46 (2017)
2. Bach, F.R., Lanckriet, G.R., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML'04 (2004)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning **3**(1), 1–122 (2011)
4. Chipman, H.A., Gu, H.: Interpretable dimension reduction. Journal of applied statistics **32**(9), 969–987 (2005)
5. Das, R., Golatkar, A., P. Awate, S.: Sparse kernel pca for outlier detection. arXiv preprint arXiv:1809.02497 (2018)
6. Dileep, A.D., Sekhar, C.C.: Representation and feature selection using multiple kernel learning. In: IJCNN 2009. pp. 717–722. IEEE (2009)
7. Fukumizu, K., Bach, F.R., Jordan, M.I.: Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. Journal of Machine Learning Research **5**(Jan), 73–99 (2004)
8. Fukumizu, K., Bach, F.R., Jordan, M.I.: Kernel dimensionality reduction for supervised learning. In: Advances in Neural Information Processing Systems. pp. 81–88 (2004)
9. Gärtner, B., Schönherr, S.: An efficient, exact, and generic quadratic programming solver for geometric optimization. In: Proceedings of the sixteenth annual symposium on Computational geometry. pp. 110–118. ACM (2000)
10. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. JMLR **12**(Jul), 2211–2268 (2011)
11. Gu, Y., Wang, Q., Wang, H., You, D., Zhang, Y.: Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery. IEEE J-STARS **8**(6), 2739–2751 (2015)

12. Jiang, W., Chung, F.l.: A trace ratio maximization approach to multiple kernel-based dimensionality reduction. Neural Networks **49**, 96–106 (2014)
13. Jolliffe, I.: Principal component analysis. In: International encyclopedia of statistical science, pp. 1094–1096. Springer (2011)
14. Kim, T.K., Kittler, J.: Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. IEEE transactions on pattern analysis and machine intelligence **27**(3), 318–327 (2005)
15. Lee, J.A., Renard, E., Bernard, G., Dupont, P., Verleysen, M.: Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation. Neurocomputing **112**, 92–108 (2013)
16. Lin, Y.Y., Liu, T.L., Fuh, C.S.: Multiple kernel learning for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(6), 1147–1160 (2011)
17. Liu, X., Wang, Z., Feng, Z., Tang, J.: A pairwise covariance-preserving projection method for dimension reduction. In: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. pp. 223–231. IEEE (2007)
18. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: Xm2vtsdb: The extended m2vts database. In: Second international conference on audio and video-based biometric person authentication. vol. 964, pp. 965–966 (1999)
19. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop. pp. 41–48. Ieee (1999)
20. Mosek, A.: The mosek optimization toolbox for matlab manual (2015)
21. Orlitsky, A., et al.: Supervised dimensionality reduction using mixture models. In: Proceedings of the 22nd international conference on Machine learning. pp. 768–775. ACM (2005)
22. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research **9**(Nov), 2491–2521 (2008)
23. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
24. Suzuki, T., Sugiyama, M.: Sufficient dimension reduction via squared-loss mutual information estimation. Neural Computation **25**(3), 725–758 (2013)
25. Tian, T.S., James, G.M.: Interpretable dimension reduction for classifying functional data. Computational Statistics & Data Analysis **57**(1), 282–296 (2013)
26. Von. Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
27. Wang, D., Tanaka, T.: Sparse kernel principal component analysis based on elastic net regularization. In: Neural Networks (IJCNN), 2016 International Joint Conference on. pp. 3703–3708. IEEE (2016)
28. Wang, Q., Gu, Y., Tuia, D.: Discriminative multiple kernel learning for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing **54**(7), 3912–3927 (2016)
29. Ye, J., Xiong, T., Janardan, R.: Cpm: A covariance-preserving projection method. In: Proceedings of the 2006 SIAM International Conference on Data Mining. pp. 24–34. SIAM (2006)