Neural Process Lett
DOI 10.1007/s11063-013-9285-x

Supervised Distance Preserving Projections

Zhanxing Zhu · Timo Similä · Francesco Corona

Abstract In this work, we consider dimensionality reduction in supervised settings and, specifically, we focus on regression problems. A novel algorithm, the supervised distance preserving projection (SDPP), is proposed. The SDPP minimizes the difference between pairwise distances among projected input covariates and distances among responses *locally*. This minimization of distance differences leads to the effect that the local geometrical structure of the low-dimensional subspace retrieved by the SDPP *mimics* that of the response space. This, not only facilitates an efficient regressor design but it also uncovers useful information for visualization. The SDPP achieves this goal by learning a linear parametric mapping and, thus, it can easily handle out-of-sample data points. For nonlinear data, a *kernelized* version of the SDPP is also derived. In addition, an intuitive extension of the SDPP is proposed to deal with classification problems. The experimental evaluation on both synthetic and real-world data sets demonstrates the effectiveness of the SDPP, showing that it performs comparably or superiorly to state-of-the-art approaches.

Keywords Dimensionality reduction · Supervised learning · Distance preservation · Regression · Classification

Z. Zhu · F. Corona (✉)
Department of Information and Computer Science, Aalto University, PO Box 15400, 00076 Aalto, Finland
e-mail: francesco.corona@aalto.fi

Z. Zhu
e-mail: zhanxing.zhu@gmail.com ·

T. Similä
Xtract Ltd, Hevosenkenkä 3, 02600 Espoo, Finland
e-mail: timo.simila@xtract.com

Published online: 09 February 2013

1 Introduction

Dimensionality reduction (DR) is an essential theme in machine learning, with specific applications for visualization, regression, classification and compression of high-dimensional data. In the past decades, the significance of DR has grown with the availability of extremely high-dimensional data, such as images, texts and gene expressions. When considered as a preprocessing step, DR helps to reduce the computational burden and to avoid the *curse of dimensionality* for most of the classification and regression models currently available.

Much research devotion on DR has focused on the unsupervised setting to handle data without labels. The most well-known methods, principal component analysis (PCA) and its nonlinear extension kernel PCA [18] learn a low-dimensional subspace by maximizing the data variance. Other important methods aim at analyzing the local geometrical structure of the data in order to learn the underlying nonlinear manifold. This approach is often referred to as *manifold learning*. Some representatives of manifold learning are locally linear embedding [17], derived from the symmetries of locally linear reconstructions, ISOMAP [21] which considers pairwise geodesic distances based on the k-nearest neighbor graph, and the Laplacian eigenmap [4] which uses an adjacency graph to formulate DR as a generalized eigenvalue problem. Another method, maximum variance unfolding [28] learns a kernel matrix by defining a neighborhood graph and retaining pairwise distances in the resulting graph. For an exhaustive discussion and comparison on unsupervised DR, we refer the interested reader to [11,25].

In supervised settings, each data sample is labelled and this information is used to guide the search of a low-dimensional space. In classification tasks, the output labels are given as discrete numbers and show which input data samples belong to the same or different classes. Supervised DR with class-labelled information is often referred to as discriminative learning. The most widely investigated method is Fisher discriminant analysis (FDA) together with its kernelized form, kernel FDA [14], and its generalized version GDA [3]. All of them try to find projection directions for a good class separation, and this is done by maximizing the ratio of between-class and within-class covariances. Some metric learning algorithms [6,29,33] search for an appropriate metric that maximizes the discriminative power in the low-dimensional subspace. In regression tasks, the labels take continuously varying real values, often referred to as output responses. Supervised DR for regression aims at finding a low-dimensional representation of the input covariates that leads to accurate predictions of the outputs. Clearly, FDA and other metric learning algorithms only consider similarities and/or dissimilarities among class labels, which limits their use only to classification tasks.

In this work, we focus on the task of supervised DR for regression. Classic partial least squares (PLS) [15,31] is a linear DR method for regression that constructs a low-dimensional subspace with orthogonal latent components by maximizing the covariance between the projected covariates and the output responses. In spirit, PLS is similar to canonical correlation analysis [9], where latent components with maximal correlation are extracted. In order to handle nonlinear projections, a kernelized version of PLS was also proposed [16]. Another approach to achieve DR for regression is sufficient dimensionality reduction (SDR) [5,12,34]. SDR finds a subspace, known as the central subspace, such that the projection leads to the independence of the output responses and the original covariates. Kernel dimension reduction (KDR) [5] is an effective method for SDR that maximizes the conditional dependence by a positive definite ordering of the expected covariance operators in the so-called probability-determining reduced kernel hilbert space. KDR does not impose particular assumptions on the underlying joint distribution of input covariates and responses, and it is a consistent estimator of the central subspace. However, the optimization of KDR is

non-convex and computationally highly demanding. Recently, Barshan et al. [2] proposed another supervised DR method called supervised principal component analysis (SPCA). Based on the Hilbert–Schmidt independence criterion [7], SPCA aims to create the principal components with maximum dependence on the output responses. SPCA is efficiently solved by an eigen-decomposition of the weighted covariance matrix enhanced by the kernel of the output responses.

In this paper, we propose a novel algorithm for supervised DR for regression named the supervised distance preserving projection (SDPP). Motivated by continuity preservation, the SDPP minimizes the difference between distances among projected covariates and distances among responses locally. Consequently, the minimization of distance differences leads to the effect that the local geometrical structure of the low-dimensional subspace mimics the geometrical characteristics of the response space. The response space can be multidimensional. The SDPP learns a linear mapping that not only facilitates an efficient regressor design, but also uncovers useful information for visualization. Being the mapping parametric, the SDPP can handle the out-of-sample data points. To deal with nonlinear problems a *kernelized* version of the SDPP (KSDPP) is derived. In addition, an extension of the SDPP for classification is suggested.

The remainder of this paper is organized as follows. Section 2 presents the SDPP, two optimization schemes to solve it, and its *kernelized* version. In Sect. 3, an experimental evaluation is conducted to show the performance of SDPP and KSDPP and compare it with state-of-the-art approaches.

2 The SDPP

The SDPP is based on simple geometric intuitions on the assumed continuity of the mapping from the covariates to the response space. The Weierstrass definition of continuity of a function states that if two points are close in the covariates space, then they are also close in the response space. Essentially, the SDPP finds a low-dimensional subspace where such a continuity is preserved. In other words, the low-dimensional space is optimized in a way that, locally, the geometry of the projected inputs *mimics* that of the output responses.

Formally, we are given n data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$ and their corresponding responses $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \in \mathbb{R}^m$, and we assume the existence of a continuous mapping $f : \mathcal{X} \mapsto \mathcal{Y}$. Provided that the input space $\mathcal{X}$ is well-sampled, we expect that for each point $\mathbf{x} \in \mathcal{X}$ and for every $\varepsilon_\mathbf{y} > 0$ there exists an $\varepsilon_\mathbf{x} > 0$ such that $d(\mathbf{x}, \mathbf{x}') < \varepsilon_\mathbf{x} \Rightarrow \delta(f(\mathbf{x}), f(\mathbf{x}')) < \varepsilon_\mathbf{y}$, where $d(\cdot, \cdot)$ and $\delta(\cdot, \cdot)$ are distance functions in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Under this condition, we want to compute a low-dimensional subspace $\mathcal{Z}$ of dimensionality r with $r \ll d$, where such a continuity is preserved. SDPP achieves this by *matching* the local geometry of the $\mathcal{Z}$ and $\mathcal{Y}$ spaces. The geometrical structure can be expressed by pairwise distances over neighborhoods of the input covariates. Inside the neighborhoods, the SDPP minimizes the difference between distances among projected covariates and distances among responses.

We assume that the subspace $\mathcal{Z}$ can be obtained by a linear transformation of $\mathcal{X}$; that is, for an input point $\mathbf{x}$, the new representation in the subspace is $\mathbf{z} = \mathbf{W}^T\mathbf{x}$, where the projection matrix $\mathbf{W} \in \mathbb{R}^{d\times r}$. Concretely, the SDPP seeks for a linear transformation $\mathbf{W}$ that parameterizes the input distances by minimizing the following criterion,

$$J(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{\mathbf{x}_j\in\mathcal{N}(\mathbf{x}_i)}(d_{ij}^2(\mathbf{W}) - \delta_{ij}^2)^2, \tag{1}$$

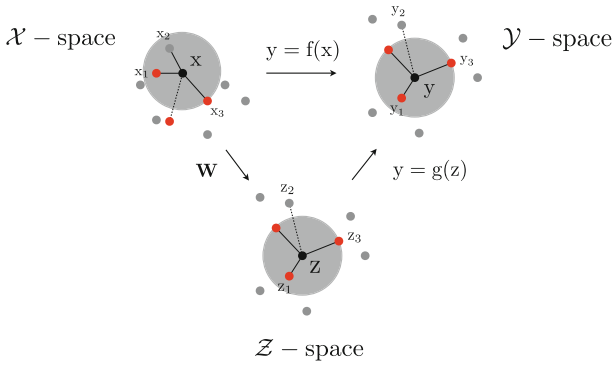

Fig. 1 An illustration of the SDPP. *Solid lines* indicate connections between nearest neighbors

where $\mathcal{N}(\mathbf{x}_i)$ is a neighborhood of $\mathbf{x}_i$. To characterize pairwise distances, we use the conventional Euclidean metric; that is $d_{ij}^2(\mathbf{W}) = \|\mathbf{z}_i - \mathbf{z}_j\|^2$ and $\delta_{ij}^2 = \|\mathbf{y}_i - \mathbf{y}_j\|^2$.

Figure 1 provides a schematic illustration of the SDPP, where, for an input point $\mathbf{x}$, three nearest neighbors of $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ are considered and a transformation $\mathbf{W}$ that leads to a similar geometry between the $\mathcal{Z}$-space and the $\mathcal{Y}$-space is found. To match the local geometry of the $\mathcal{Y}$-space, one of the three nearest neighbors, $\mathbf{x}_2$, is moved, after projection, outside the neighborhood in the $\mathcal{Z}$-space while another point is moved inside. This match is beneficial to the regression from the subspace $\mathcal{Z}$ to the response space $\mathcal{Y}$ and to the visualization for revealing the relationship between the inputs and their responses.

The criterion of the SDPP, in Eq. 1, is similar in formulation of the S-Stress [20], the objective used in one of the variants of the multidimensional scaling (MDS). Both the SDPP and the S-Stress are, in fact, formulated from *squared* pairwise distances, whereas the cost function of other MDS methods, the conventional Kruskal's stress, is defined using pairwise distances. However, three main differences between the SDPP and the S-Stress can be observed. Firstly, the goal of SDPP is to produce a faithful low-dimensional representation for the purpose of regression, thus it is specifically designed to address the DR problem in a supervised setting. Consequently, squared pairwise distances in the response space are considered. On the contrary, the S-Stress is defined for an unsupervised method and, thus no response information is used. Secondly, in order to capture the local geometry of the data, the SDPP incorporates a neighborhood graph into the cost, whereas the S-Stress only attempts to achieve global distance preservation. As pointed out by Groenen and van de Velden [8], the disadvantage of the S-Stress is that it tends to give solutions in which large dissimilarities are overemphasized and the small dissimilarities are not well represented. Thirdly, the SDPP produces a linear parametric mapping, whereas most variants of MDS produce a point-wise mapping. Parametric MDS has been investigated by Webb [27] and Tipping and Lowe [22] but, in comparison to the SDPP, with different optimization methods and a slightly different formulation of the cost function. Webb uses an iterative majorization while Tipping and Lowe propose a two-step procedure which is reminiscent to the EM philosophy to likelihood maximization.

2.1 Optimization of the SDPP

To optimize the objective function of the SDPP, we consider two different strategies. Specifically, we discuss a semidefinite quadratic linear programming (SQLP) and conjugate-gradient (CG) optimization.

2.1.1 SQLP

Starting from the square of the pairwise distances in the $\mathcal{Z}$-space, $d_{ij}^2(\mathbf{W}) = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)$ with $\mathbf{P} = \mathbf{W}\mathbf{W}^T$ a positive semidefinite (PSD) matrix denoted as $\mathbf{P} \succeq 0$, we formulate the optimization problem of SDPP as an instance of convex quadratic semidefinite programming (QSDP).

For notational simplicity, we denote $\boldsymbol{\tau}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. The squared pairwise distances in the input space, parameterized by the linear model $\mathbf{W}$, can thus be expressed as

$$d_{ij}^2(\mathbf{W}) = \boldsymbol{\tau}_{ij}^T\mathbf{P}\boldsymbol{\tau}_{ij} = \text{vec}(\boldsymbol{\tau}_{ij}\boldsymbol{\tau}_{ij}^T)^T\text{vec}(\mathbf{P}) = \mathbf{l}_{ij}^T\mathbf{p},$$

where vector $\mathbf{l}_{ij} = \text{vec}(\boldsymbol{\tau}_{ij}\boldsymbol{\tau}_{ij}^T)$, vector $\mathbf{p} = \text{vec}(\mathbf{P})$, with $\text{vec}(\cdot)$ an operator that concatenates all the columns of a matrix into a new vector. Based on this manipulation, we can rewrite the objective as a function of $\mathbf{p}$,

$$\begin{aligned} J(\mathbf{p}) &= \mathbf{p}^T \underbrace{\left(\frac{1}{n}\sum_{ij}\mathbf{G}_{ij}\mathbf{l}_{ij}\mathbf{l}_{ij}^T\right)}_{\mathbf{A}}\mathbf{p} + \underbrace{\left(-\frac{2}{n}\sum_{ij}\mathbf{G}_{ij}\delta_{ij}^2\mathbf{l}_{ij}\right)^T}_{\mathbf{b}}\mathbf{p} \\ &\quad + \underbrace{\frac{1}{n}\sum_{ij}\mathbf{G}_{ij}\delta_{ij}^4}_{c} \\ &= \mathbf{p}^T\mathbf{A}\mathbf{p} + \mathbf{b}^T\mathbf{p} + c, \end{aligned} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{d^2\times d^2}$, $\mathbf{b} \in \mathbb{R}^{d^2\times 1}$, and c is a constant that can be ignored later in optimization. In Eq. 2, $\mathbf{G}_{ij}$ is the neighborhood graph of $\mathbf{x}_i$ defined as

$$\mathbf{G}_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_j \text{ is a neighbor of } \mathbf{x}_i, \\ 0 & \text{otherwise.} \end{cases}$$

The optimization of SDPP is thus obtained by solving the equivalent QSDP problem,

$$\begin{aligned} \min_{\mathbf{p}} \quad & \mathbf{p}^T\mathbf{A}\mathbf{p} + \mathbf{b}^T\mathbf{p} \\ \text{s.t.} \quad & \mathbf{P} \succeq 0 \end{aligned} \tag{3}$$

It is important to notice that the QSDP formulation does not optimize the projection matrix $\mathbf{W}$ directly, instead it optimizes the PSD matrix $\mathbf{P} = \mathbf{W}\mathbf{W}^T$. The projection matrix $\mathbf{W}$ can be computed as the square root of $\mathbf{P}$ or, alternatively, one could apply a singular value decomposition (SVD) on $\mathbf{P}$ to obtain an orthogonal projection matrix $\mathbf{W}$. In the latter case, the ith column of $\mathbf{W}$ is calculated as $\sqrt{\lambda_i}\mathbf{v}_i$, being λ_i and $\mathbf{v}_i$ the ith eigenvalue and eigenvector of $\mathbf{P}$, respectively. The dimensionality of the projection subspace is determined by analyzing the eigenvalues.

The form in Eq. 3 is often re-written as a semidefinite programming (SDP) problem, see [13,19,30]. In this paper, we take advantage of the low-rank structure of $\mathbf{A}$ and we apply the

idea from [32] that reformulates the QSDP as the equivalent SQLP problem

$$\begin{aligned} \min_{\mathbf{p},\mathbf{u}} \quad & (\mathbf{e}_1 - \mathbf{e}_2)^T \mathbf{u} + \mathbf{b}^T \mathbf{p} \\ \text{s.t.} \quad & (\mathbf{e}_1 + \mathbf{e}_2)^T \mathbf{u} = 1, \\ & \mathbf{B}\mathbf{p} - \mathbf{C}\mathbf{u} = 0, \\ & \mathbf{u} \in \mathbb{K}_{q+2}, \\ & \mathbf{P} \succeq 0, \end{aligned} \tag{4}$$

where q is the rank of $\mathbf{A}$, by Cholesky factorization $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ with $\mathbf{B} \in \mathbb{R}^{q \times d^2}$, $\mathbf{C} = [\mathbf{0}_{q\times 2}, \mathbf{I}_{q\times q}]$, $\mathbf{e}_i$ is the ith basis vector with $i = 1, 2, \ldots, q+2$ and $\mathbb{K}_m$ is the second-order cone of dimension m (i.e., $\mathbb{K}_m = \{(x_0; \mathbf{x}) \in \mathbb{R}^m | x_0 \geq \|\mathbf{x}\|\}$).

The SQLP formulation sets the optimization problem into a standard framework of optimization with semidefinite constraints, which is supported by many efficient optimization libraries. The standard Matlab toolbox SDPT3-4.0 [23,24] can be used for solving it efficiently. The solution of the SQLP problem requires $O(d^{6.5})$ arithmetic operations and it is, in that sense, convenient when compared to the SDP solution that requires $O(d^9)$ operations. However, due to the large size of the matrix $\mathbf{A}$, we suggest to consider it only for not very high-dimensional problems (e.g. $d < 100$).

2.1.2 CG Optimization

When the dimension of original data set is very high, the size of $\mathbf{A}$ in the SQLP formulation becomes extremely large. This aspect might bring practical limitations related to storing capacity and further optimization even shielded by efficient optimization routines. To overcome these shortcomings, we designed an alternative optimization approach based on the conventional CG search.

After denoting the (squared) pairwise distances as $\mathbf{D}_{ij} = d^2_{ij}(\mathbf{W})$ and $\boldsymbol{\Delta}_{ij} = \delta^2_{ij}$ and, remembering that $\mathbf{G}_{ij}$ is the neighborhood graph of $\mathbf{x}_i$, the objective function of SDPP in Eq. 1 can be re-written in the equivalent form,

$$J(\mathbf{W}) = \frac{1}{n} \sum_{ij} \mathbf{G}_{ij} (\mathbf{D}_{ij} - \boldsymbol{\Delta}_{ij})^2. \tag{5}$$

The gradient of the objective function with respect to $\mathbf{W}$ is thus derived as equal to

$$\nabla_{\mathbf{W}} J = \frac{4}{n} \sum_{ij} \mathbf{G}_{ij} (\mathbf{D}_{ij} - \boldsymbol{\Delta}_{ij}) \boldsymbol{\tau}_{ij} \boldsymbol{\tau}_{ij}^T \mathbf{W}. \tag{6}$$

A more compact form of the gradient in Eq. 6 can be obtained after denoting $\mathbf{Q} = \mathbf{G} \odot (\mathbf{D} - \boldsymbol{\Delta})$ with $\odot$ representing the element-wise product of two matrices, the symmetric matrix $\mathbf{R} = \mathbf{Q} + \mathbf{Q}^T$, and $\mathbf{S}$ a diagonal matrix with $\mathbf{S}_{ii} = \sum_j \mathbf{R}_{ij}$. The algebraic manipulations reported in Appendix A lead to a matrix-form of the gradient,

$$\nabla_{\mathbf{W}} J = \frac{4}{n} \mathbf{X}^T (\mathbf{S} - \mathbf{R}) \mathbf{X} \mathbf{W}, \tag{7}$$

where each row of the data matrix $\mathbf{X}$ is a point $\mathbf{x}_i$. $\mathbf{L} = \mathbf{S} - \mathbf{R}$ is the Laplacian matrix.

The CG optimization of the objective function of the SDPP is sketched in Algorithm 1. It is worthwhile noticing that using the CG approach allows for a direct optimization of the projection matrix $\mathbf{W}$. In comparison to the SQLP approach where the dimensionality of the projection subspace is selected in a post-processing step, here it is defined beforehand.

Algorithm 1 CG optimization of the SDPP

Input: training data matrix $\mathbf{X}$ and $\mathbf{Y}$, neighborhood graph $\mathbf{G}$, initialized projection matrix $\mathbf{W}_0$
Output: optimized projection matrix $\mathbf{W}$.

1. Compute gradient $\nabla_{\mathbf{W}} J$;
2. Vectorize the projection matrix, $\mathbf{w}_0 = vec(\mathbf{W}_0)$;
3. Vectorize the gradient, $\mathbf{g}_0 = vec(\nabla_{\mathbf{W}} J)$;
4. Initialize the conjugate direction as $\boldsymbol{\nu}_0 = -\mathbf{g}_0$;
for $t = 1 \rightarrow T$ **do**
  5. Calculate β_t by Polak-Ribiére's rule, $\beta_t = \frac{\mathbf{g}_t^T(\mathbf{g}_t - \mathbf{g}_{t-1})}{\mathbf{g}_{t-1}^T \mathbf{g}_{t-1}}$;
  6. Update the conjugate direction, $\boldsymbol{\nu}_t = -\mathbf{g}_t + \beta_t \boldsymbol{\nu}_{t-1}$;
  7. Perform line search, $\eta_t = \arg\min_\eta \ J(\mathbf{w} + \eta \boldsymbol{\nu}_t)$;
  8. Update $\mathbf{w}$, $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta_t \boldsymbol{\nu}_t$
end for
9. Reshape the vector $\mathbf{w}_{T+1}$ into the matrix $\mathbf{W}$.

2.2 Kernel Extension of the SDPP (KSDPP)

In general, the performance of supervised dimension reduction by linear projection might degrade in the presence of nonlinearly distributed data. In this section, we present a straightforward extension of the SDPP to handle nonlinear problems, the KSDPP. Based on the intuition behind the so-called kernel trick, KSDPP first re-maps the data from the original input space to another higher (even infinite) dimensional feature space, $\boldsymbol{\phi} : \mathcal{X} \mapsto \mathcal{H}$, and then performs its linear projection in this new feature space.

Denote the element of the kernel matrix $\mathbf{K}$ as $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$. Based on representation theorem [10], we assume that the projection matrix is written as $\mathbf{W} = \boldsymbol{\Phi}\boldsymbol{\Omega}$ with $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]$, we can express the squared distances in the reduced feature space, now parameterized by the linear transformation $\boldsymbol{\Omega}$, as

$$\begin{aligned} d_{ij}^2(\boldsymbol{\Omega}) &= (\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j))^T \mathbf{W}\mathbf{W}^T (\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)) \\ &= (\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j))^T \boldsymbol{\Phi}\boldsymbol{\Omega}\boldsymbol{\Omega}^T \boldsymbol{\Phi}^T (\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)) \\ &= (\boldsymbol{\Phi}^T \boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\Phi}^T \boldsymbol{\phi}(\mathbf{x}_j))^T \boldsymbol{\Omega}\boldsymbol{\Omega}^T (\boldsymbol{\Phi}^T \boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\Phi}^T \boldsymbol{\phi}(\mathbf{x}_j)) \\ &= (\mathbf{K}_i - \mathbf{K}_j)^T \boldsymbol{\Pi} (\mathbf{K}_i - \mathbf{K}_j), \end{aligned}$$

where the PSD matrix $\boldsymbol{\Pi} = \boldsymbol{\Omega}\boldsymbol{\Omega}^T$ with $\boldsymbol{\Pi} \in \mathbb{R}^{n \times n}$ and $\mathbf{K}_i$ is the ith column of $\mathbf{K}$. Note that the squared distances of the responses in the feature space are now expressed as

$$\delta_{ij}^2 = \mathbf{K}_{ii}^y + \mathbf{K}_{jj}^y - 2\mathbf{K}_{ij}^y,$$

where $\mathbf{K}^y$ is the kernel matrix for the response space. After replacing $\mathbf{X}$ and $\mathbf{W}$ with $\mathbf{K}$ and $\boldsymbol{\Omega}$, respectively, the optimization of KSDPP is same as that of the original SDPP. Moreover, the out-of-sample projection under KSDPP is $\mathbf{z} = \boldsymbol{\Omega}^T \mathbf{K}_{\mathbf{x}}$.

In application areas like image processing and genomics, the dimensionality d of the data is often much larger than the number n of data points. In these situations, applying SDPP directly might be impractical due to a large PSD matrix with size $d \times d$. Favorably, an

additional benefit of KSDPP is that it can reduce the computational complexity by learning a much smaller projection matrix $\boldsymbol{\Omega}$, with size $n \times n$.

2.3 A Continuity Measure for Neighborhood Selection

In the definition of the SDPP, we have defined locality for each data point $\mathbf{x}_i$ by considering its k nearest neighbors $\mathcal{N}(\mathbf{x}_i)$. The number of neighbors k is a hyper-parameter to be provided by the user beforehand or to be tuned directly from data. Remembering that SDPP tries to enforce the local continuity of the mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ also after projection onto the $\mathcal{Z}$-space, we introduce a consistent heuristic that assists the user when selecting the size k of the neighborhood. The selection is based on a continuity measure of the mapping $g : \mathcal{Z} \mapsto \mathcal{Y}$.

Let $V_{k_r}(i)$ be the set of those data points that are in the neighborhood of size k_r in the $\mathcal{Z}$-space but not in the $\mathcal{Y}$-space, and let $r(i, j)$ be the rank of y_j in the ordering based on its distance from y_i. In the spirit of [26], we measure the continuity of g as

$$M_{\text{cont}}(k_r) = 1 - C(k_r) \sum_{i=1}^{n} \sum_{j \in V_{k_r}(i)} (r(i, j) - k), \tag{8}$$

where the scaling term is defined as

$$C(k_r) = \begin{cases} \frac{2}{nk_r(2n-3k_r-1)} & \text{if } k_r < \frac{n}{2}, \\ \frac{2}{n(n-k_r)(n-k_r-1)} & \text{if } k_r \geq \frac{n}{2}. \end{cases}$$

Practically, we firstly learn different SDPPs using different neighborhood sizes k in order to obtain different low-dimensional representations. Then, we calculate for each low-dimensional representation the corresponding continuity measures for a sequence of neighborhoods k_r. For the sequence of k_r values, we select the optimal size k of $\mathcal{N}(\mathbf{x}_i)$ as the one with jointly highest continuities.

3 Experimental Evaluation

In this section, we illustrate the effectiveness of the SDPP in a number of regression tasks, and we compare its performance with several state-of-the-art methods for supervised DR. For comparison, we consider PLS, SPCA and KDR. In addition, we present and discuss a simple extension of SDPP that renders it suitable also for classification tasks. In this case, also FDA is considered for comparison.

In general, when the KSDPP is used, standard RBF Gaussian kernels are employed. The same applies also for SPCA, kernel SPCA and KDR that are kernel-methods by definition. When appropriate, a kernelized version of PLS (KPLS) is used for consistency. The optimal kernel width is estimated by cross-validation based on a set of candidate kernel parameters. For the neighborhood size selection in SDPP and KSDPP, we always use the heuristic based on continuity measure presented in Eq. 8.

The analysis is based on both synthetic and real-world problems. Table 1 presents a brief description of the data sets used for the experimental evaluation.

Table 1 Description of data sets for experimental evaluation

Data sets	Dimensionality of $\mathbf{x}$	Dimensionality of $\mathbf{z}$	Size of training set	Size of test set
Linear data	5	1	500	500
"Parity"	5	2	500	500
Curve line	5	1	500	500
Servo	4	1–4	111	56
Tecator-fat	100	1–4	143	72
Auto-price	15	1–4	106	53
Ducks	4, 096	1	62	10
Tai Chi	5	2	500	1,500
Glass	10	2	143	71

The first seven data sets refer to regression problems and the last two to classification problems

3.1 The SDPP for Regression

3.1.1 Synthetic Data Sets

We consider three synthetic data sets for evaluating the performance of SDPP and we compare its performance against PLS, SPCA and KDR. We assume that n points are available for training, $\{\mathbf{x}_i\}_{i=1}^n$ and each data point has features $\mathbf{x} = [X_1, \ldots, X_d]^T$.

In the first two data sets, we generate 1,000 five-dimensional data points with uniform distribution, $\mathbf{x} \in U[0, 1]^5$. Only the first two features are used for building the response y. The first half of the points is used for training and the remaining for test.

Linear data The responses are constructed according to the simple linear model,

$$y = 2X_1 + 3X_2 + \varepsilon,$$

where the noise term $\varepsilon \sim N(0, 0.5^2)$. Figure 2a shows the plane defined by the output responses of the test data with respect to two effective features X_1 and X_2. Combining X_1 and X_2 with their corresponding coefficients, we also plot the true projection of the test set onto a one-dimensional space, see Fig. 2b.

For all the four methods, a mono-dimensional projection is learned. For SDPP, the optimal projection matrix (vector) $\mathbf{W} \in \mathbb{R}^{5\times 1}$ is selected after measuring the continuity heuristic on the training data to determine the suitable neighborhood size. From Fig. 3a, where continuity measures with respect to different k and k_r are plotted, it is possible to observe that a suitable value for k should be in the range [32, 64]. The learned $\mathbf{W}$ is then used to project the test points, Fig. 2c. Figure 2d–f presents the projection results obtained with PLS, SPCA and KDR. In the figures, the x axis and y axis represent the projected covariates and their actual output responses. Each point in the plots is dyed according to its true response value y, where warm (red) and cold (blue) color depict high and low values, respectively. For this simple linear problem, all the four methods are able to find a correct projection and no clear differences can be detected. This is further confirmed in Fig. 3b, where continuity measures with respect to different values of k_r are plotted for all the methods.

Smoothed parity The responses are constructed according to the nonlinear model

$$y = \sin(2\pi X_1)\sin(2\pi X_2) + \varepsilon,$$

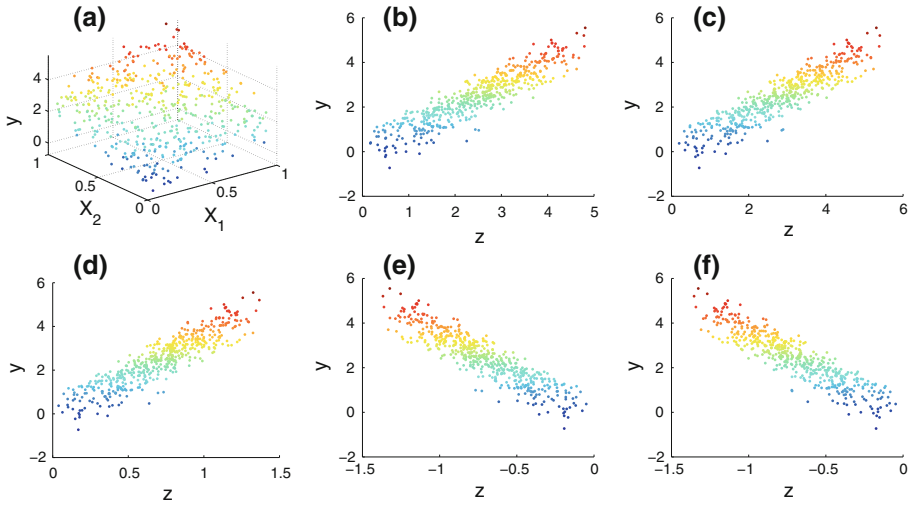

Fig. 2 Linear data: **a** 3D plot of test points with two effective features X_1 and X_2, **b** true projection of test data points in one-dimensional space, **c** SDPP projection, **d** PLS projection, **e** SPCA projection, and **f** KDR projection

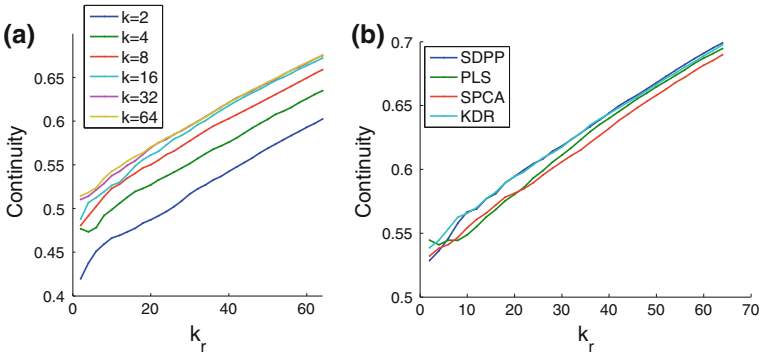

Fig. 3 Linear data: continuity measures. **a** The SDPP: continuities with respect to different k and k_r on the training set and **b** continuities from the four methods on the training set

where the noise term $\varepsilon \sim N(0, 0.1^2)$. Figure 4a shows the output response of the *smoothed parity* function with respect to the two effective features X_1 and X_2. Clearly, after DR, only the first two features should be preserved as much as possible. The true two-dimensional projection is plotted in Fig. 4b.

For all the four methods, a bi-dimensional projection is thus learned. In the learning phase of SDPP, the suitable neighborhood size is in the range [8,16] based on Fig. 5a. And the bi-dimensional projection result for test points is presented in Fig. 4c, where the two effective projection directions are detected correctly. Figure 4d–f shows the bi-dimenional projection for the test set by PLS, SPCA and KDR. In this case, PLS and SPCA are clearly uncapable of finding the first two informative features but, on the other hand, KDR obtains similar results to SDPP. However, KDR requires a much longer time for training: 65.77 s for 50 iterations compared with only 2.91 s training time of SDPP, on a standard workstation with 2.83 GHz CPU and 4 GB RAM running Matlab R2011a. The high computational burden is mainly

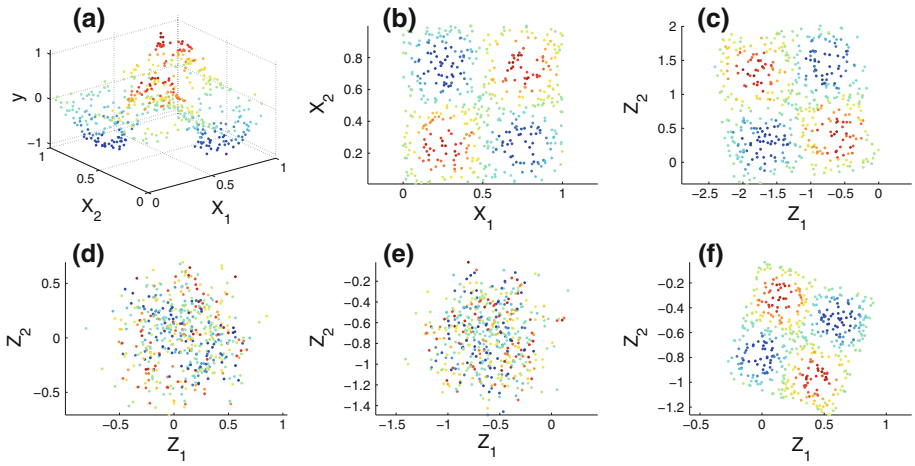

Fig. 4 *Smoothed parity*: **a** 3D plot of test points with two effective features X_1 and X_2, **b** true projection of test data points in two-dimensional space, **c** SDPP projection, **d** PLS projection, **e** SPCA projection, and **f** KDR projection

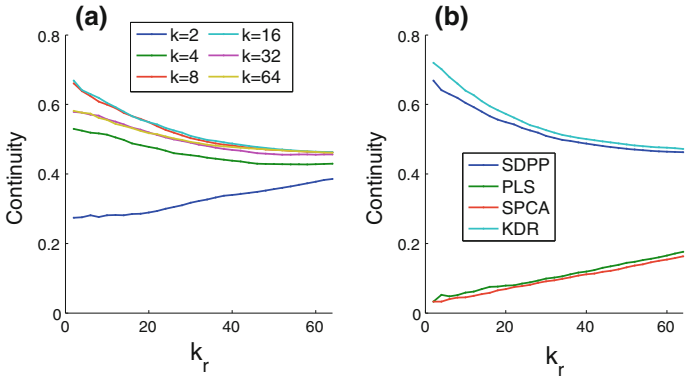

Fig. 5 *Smoothed parity*: continuity measures. **a** The SDPP: continuities with respect to different k and k_r on the training set and **b** continuities from the four methods on the training set

due to the fact that KDR needs to compute the inverse of the kernel matrix in each gradient descent step. Additionally, SDPP and KDR provide the highest continuities over the whole range of k_r values, see Fig. 5b. These results are consistent with the correct projection results obtained by the two methods and show the moderate superiority of KDR over SDPP on this problem.

Curved line The response is constructed by a latent variable t uniformly distributed in $[0, 4\pi]$, according to the following model:

$$\mathbf{x} = [\cos(t), \sin(t), 0.01t, \varepsilon_{\mathbf{x}}]^T$$
$$y = t + \varepsilon_y,$$

where $\varepsilon_{\mathbf{x}}$ is 2-dimensional vector of *noisy* input variables, $\varepsilon_{\mathbf{x}} \sim U[0, 1]^2$, and ε_y is the noise term for response, $\varepsilon_y \sim N(0, 1)$. Also for this problem, 1,000 data point are generated and the first half is used for learning the projection models while the remaining is used for testing.

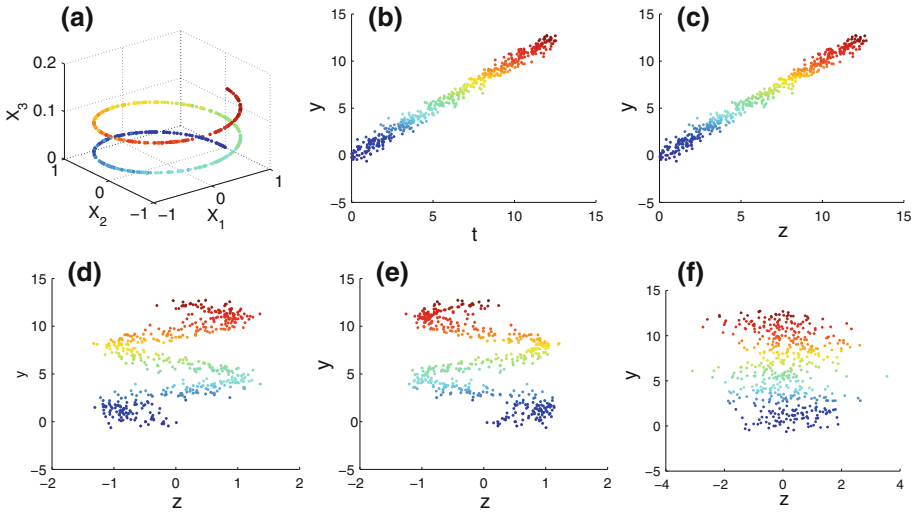

Fig. 6 Curved line: **a** 3D plot of test points with three effective features X_1, X_2 and X_3; **b** true projection of test data points in one-dimensional space, **c** SDPP projection, **d** PLS projection, **e** SPCA projection, and **f** KDR projection

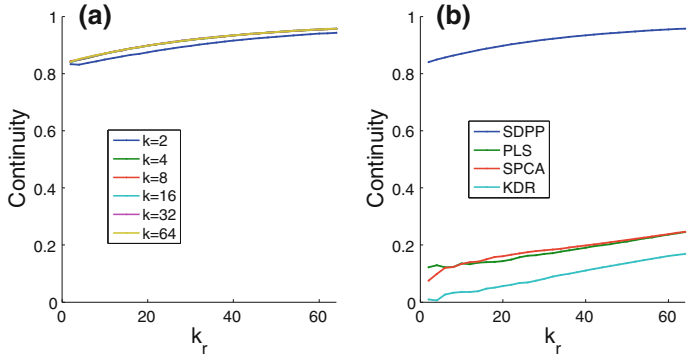

Fig. 7 Curved line: continuity measures. **a** SDPP: continuities with respect to different k and k_r on the training set and **b** continuities from the four methods on the training set

Figure 6a shows a plot of the dyed responses as a function of the three useful features. Notice that the true manifold structure should be determined only by the latent variable t along a straight line, Fig. 6b.

Figure 6c–f shows the results for the one-dimensional projection of the test set obtained by the four methods. In the plots, the x axis describes the projected variable Z and its true response is represented along the y axis. Again, SDPP is capable to successfully detect the third feature as the one that is linearly associated with the response. On the other hand, PLS, SPCA and KDR do not seem to provide faithful projections for this data set. From Fig. 7, it is possible to notice the very high continuity measures obtained by the SDPP, both in an absolute sense and when compared to the other methods. An additional confirmation of this result can be seen from the projection matrices learned by the four methods:

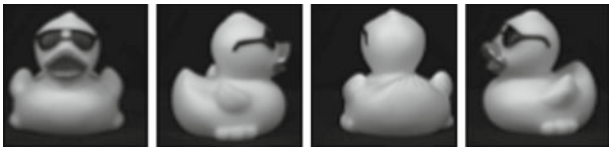

Fig. 8 The rotating duck data set: four 64 × 64 input images

$$\begin{aligned}
\mathbf{W}_{SDPP} &= [-0.04,\ -0.08,\ \underline{100.68},\ -0.03,\ 0.03]^T,\\
\mathbf{W}_{PLS} &= [-0.13,\ \underline{0.97},\ 0.13,\ 0.06,\ 0.15]^T,\\
\mathbf{W}_{SPCA} &= [0.16,\ \underline{0.98},\ -0.04,\ 0,\ 0.11]^T,\\
\mathbf{W}_{KDR} &= [-0.20,\ -0.36,\ -0.02,\ \underline{-0.88},\ -0.23]^T.
\end{aligned}$$

3.1.2 Real-World Data Sets

In this section, we analyze the behavior of the SDPP and KDSPP on a number of real-world data sets and we compare its performance against other methods. Two main problems are considered, (i) the prediction of the rotation angles of an object portrayed on an image and, (ii) the prediction accuracy of a simple linear regression model on a selection of data sets from the UCI repository [1].

Predicting rotation angles from image manifolds The data set consists of a collection of images of a rubber duck in different rotation angles, Fig. 8. The images are selected from the Amsterdam Library of Object Images (ALOI)[1] image database. The goal is to embed the high-dimensional images into a low-dimensional subspace that unveils this aspect of image variations, and to predict the rotation angles of the ducks given for a set test images. The set contains 72 images with uniformly distributed rotation angles, $[0°, 5°, \ldots, 355°]$. We use ten images with equally spaced rotation angles for testing the projection models learned from the remaining 62 training images.

In our experiment, we resized the original images and cropped them into a suitable size of 64 × 64 pixels. In this case, the dimensionality of the each image (4,096) is still much larger than the number of data points available for learning (62). In order to further reduce the computational complexity of the solution, we thus use KSDPP and compare its performances against kernel PLS, kernel SPCA and KDR.

Figure 9 shows the results obtained by the four aforementioned methods when projecting the input images onto a mono-dimensional space. In the plots, the x axis represents the one-dimensional embedding and the y axis shows the corresponding rotation angle. Small and large dots are used to indicate the training and test images, respectively, and the test images are also marked with their true rotational angle. Based on the experimental results, we can observe how KSDPP successfully unfolds the image manifold and achieves a representation where simple regression models, even a linear model, might be sufficient for predicting the rotation angles of test ducks. On the other hand, the other three methods do not provide useful embeddings in the sense that one embedded input point typically corresponds to different rotation angles. Clearly, this shortcoming violates the surjectivity of the regression function. Such a violation leads to obvious difficulties or even impossibility for regression from the obtained low-dimensional space.

Regression on UCI data sets Three commonly used data sets from the UCI repository (namely, Servo, Tecator-fat and Auto-price) are considered for experimental evaluation. The

[1] http://staff.science.uva.nl/~aloi/.

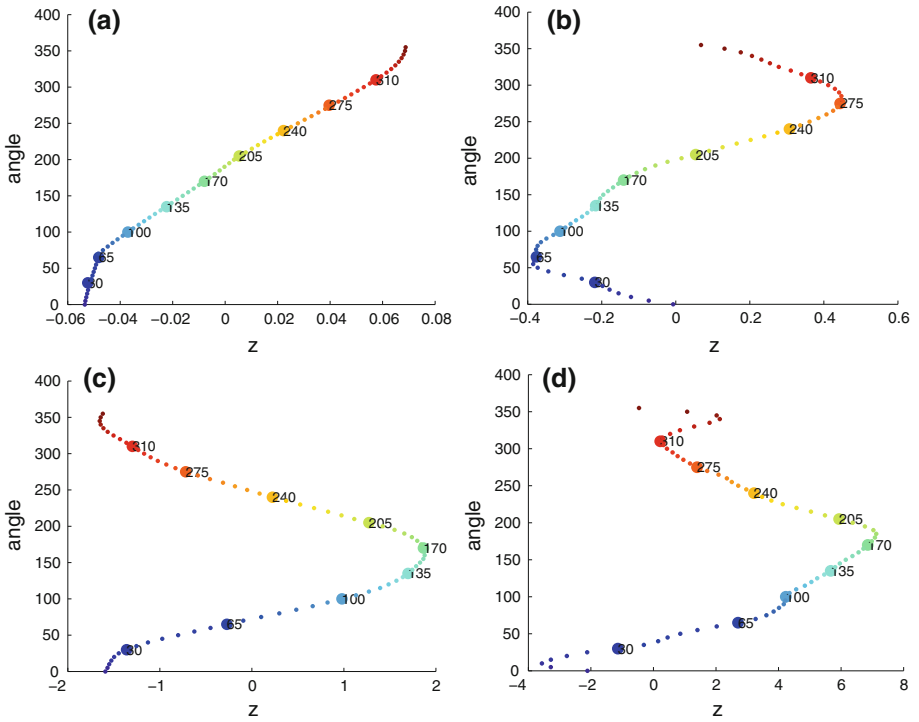

Fig. 9 The rotating duck data set: one-dimensional embeddings: **a** KSDPP embedding, **b** KPLS embedding, **c** KSPCA embedding, and **d** KDR embedding

description of the data sets is summarized in Table 1. All the sets are preprocessed by mean centering and normalized to unit variance. We randomly split the each data set into two parts, two thirds for training and the remaining for test. After the DR step, a simple linear model is used for regression.

Table 2 reports the regression accuracy in terms of *root mean squared error* RMSE ("mean $\pm$ std") on test sets, for each DR method after projecting onto low-dimensional subspaces of different dimensionality r. For each dimensionality, bold numbers are used to indicate the lowest RMSE. We can observe that, in most cases, SDPP exhibits an equivalent or better generalization performance compared with the other three methods. Advantageously, the performance of SDPP does not change much with the projection dimensionality, and accurate regression models are achieved for small values of r. This aspect is mostly beneficial for practical applications, where the projection dimensionality is often determined by cross-validation. Moreover, SDPP is more stable regarding the random splitting of the data sets since the standard deviation of the RMSE is smaller than that of the other methods.

3.2 SDPP for Classification

In its basic formulation, the SDPP criterion is based on the dissimilarity between two output responses δ_{ij} in terms of their Euclidean distance, Eq. 1. Such a formulation is specifically derived on considerations on a typical regression task. In order to extend SDPP to classification tasks, the aforementioned dissimilarity needs to be redefined to represent differences

Table 2 Three UCI data sets: prediction results as RMSE ("mean ± std") on test

Data sets	r	SDPP	PLS	SPCA	KDR
Servo	1	1.2180 ± 0.0748	1.3555 ± 0.1194	1.3804 ± 0.1194	$\mathbf{1.1946 \pm 0.1083}$
	2	$\mathbf{1.1730 \pm 0.1140}$	1.2432 ± 0.1568	1.3622 ± 0.1495	$\mathbf{1.1625 \pm 0.1475}$
	3	1.1699 ± 0.1344	1.1719 ± 0.1305	1.3301 ± 0.1581	$\mathbf{1.1592 \pm 0.1179}$
	4	$\mathbf{1.1733 \pm 0.1315}$	$\mathbf{1.1733 \pm 0.1315}$	$\mathbf{1.1733 \pm 0.1315}$	1.6181 ± 0.1427
Tecator-fat	1	$\mathbf{2.2650 \pm 0.3943}$	6.9072 ± 0.5501	7.2070 ± 0.5689	5.4043 ± 2.1468
	2	$\mathbf{2.2529 \pm 0.3286}$	5.4518 ± 0.4608	5.7623 ± 0.4530	4.2125 ± 1.6056
	3	2.2348 ± 0.1870	2.3142 ± 0.1795	4.7267 ± 0.9667	$\mathbf{2.2020 \pm 0.1645}$
	4	$\mathbf{2.2061 \pm 0.1472}$	2.2071 ± 0.2393	$\mathbf{2.1791 \pm 0.2192}$	2.2068 ± 0.2812
Auto-price	1	2.8772 ± 0.3536	2.8733 ± 0.3405	2.9272 ± 0.3173	$\mathbf{2.7282 \pm 0.3909}$
	2	$\mathbf{2.6987 \pm 0.3750}$	2.7205 ± 0.3912	2.9043 ± 0.2737	2.7680 ± 0.4145
	3	$\mathbf{2.6757 \pm 0.3728}$	2.6978 ± 0.3941	2.7348 ± 0.3399	2.8321 ± 0.3362
	4	$\mathbf{2.6955 \pm 0.3753}$	2.7033 ± 0.3823	$\mathbf{2.6901 \pm 0.3937}$	2.7853 ± 0.3937

between classes. A simple reformulation that sets $\delta_{ij} = 0$ when two points belong to the same class and $\delta_{ij} = 1$ when two points belong to different classes allows for a direct use of the SDPP as DR method also for classification.

3.2.1 Tai Chi

Figure 10a presents the well-known Tai Chi symbol in the Asian culture. The black and white regions indicate two opposing entities, Yin and Yang, respectively. The concepts of Yin and Yang provide the intellectual framework of the scientific development in ancient China, especially in fields like biology and traditional medical sciences. The basic structure of Tai Chi is formed by drawing one large circle, two medium half-circles and two small circles. The two small Yin and Yang circles, located at the centers of the Yang and Yin half-circles that are tangent to each other and also to the large circle. In the following, we use the Tai Chi symbol to define a toy classification problem, Fig. 10b, where Yin and Yang are two distinct classes.

For the task, a discretized version of the symbol is simulated as follows:

- X_1 and X_2 distribute uniformly within the large circle. We then assign the class label $y = +1$ and $y = -1$ to the points in the Yin area and Yang area, respectively.
- X_3, X_4, X_5 are noisy features following the distribution $N(0, \mathbf{I}_3)$.

Thus, the original input $\mathbf{x}$ is a five-dimensional vector. The goal is to identify the first two effective directions for a correct classification of Yin and Yang. We generated 2000 points and used the first 500 for training and the remaining for test.

Four methods, the SDPP, FDA, SPCA and KDR are considered for comparison. Figure 10 presents the two-dimensional projections for the test set. Clearly, the SDPP finds the two projection directions successfully while the other methods tend to mix the two classes in the areas where the two small Yin and Yang circles are located.

3.2.2 Visualization on Glass Data

Now we apply the SDPP to analyze the Glass data set for classification, also from the UCI repository. To demonstrate its effectiveness on visualization for classification, we randomly select two thirds of the data for learning the low-dimensional projection subspace and the

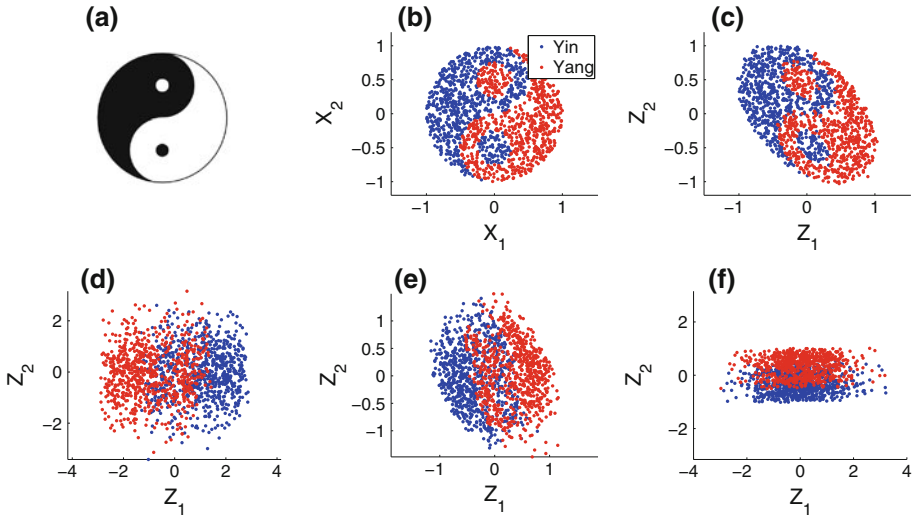

Fig. 10 The Tai Chi: **a** the original Tai Chi, **b** simulation of Tai Chi model, **c** SDPP projection, **d** FDA projection, **e** SPCA projection, and **f** KDR projection

remainder for testing the results. Again, the SDPP, FDA, SPCA and KDR are considered for comparison.

Figure 11 shows the projections obtained by the four methods. From the panels, it is possible to notice that SDPP yields an accurate projection, with all the classes clearly separated. All the test points are projected along a straight line, thus indicating that only one-dimensional projection might be sufficient for correctly visualizing the data. Such an intuitive representation cannot be achieved by the other methods. Furthermore FDA, SPCA and KDR do not appear to be successful in fully separating the classes, as observable by noticing the points overlapping between them.

4 Conclusion

This work presents a novel algorithm for supervised DR for regression, the SDPP. Motivated by continuity preservation, the SDPP minimizes the difference between distances among projected covariates and distances among responses locally. The minimization drives the SDPP toward the definition of a low-dimensional subspace where the local geometrical structure of the input data mimics the geometrical characteristics of the response data. The learned subspace not only facilitates an efficient regressor design, but it also uncovers useful information for visualization. The SDPP learns such an embedding through a linear mapping and thus it can easily handle the out-of-sample data. For solving the SDPP efficiently, two optimization schemes, SQLP and CG optimization, have been introduced. We also derived KSDPP, to deal with nonlinear data. The KSDPP also reduces the computational complexity of the basic SDPP formulation when the dimensionality of the input covariates is much larger than the number of data samples. In addition, an intuitive extension of SDPP for classification is suggested.

A number of synthetic and real-world data sets are considered for the experimental evaluation and comparison with state-of-the-art methods. Based on the experimental results,

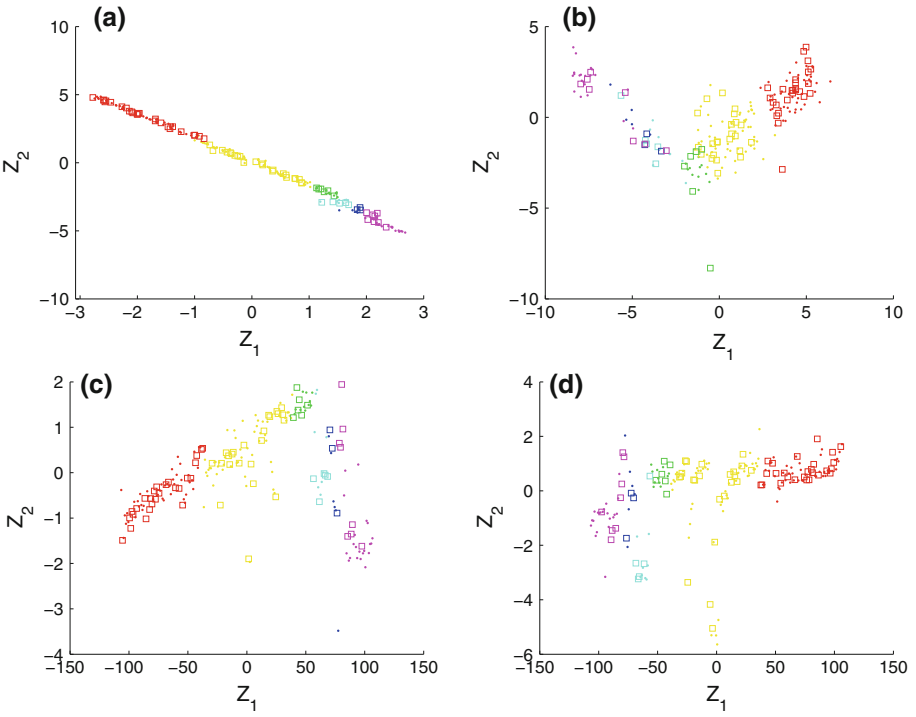

Fig. 11 The Glass data set: two-dimensional projections: **a** SDPP projection, **b** FDA projection, **c** SPCA projection, **d** KDR projection. *Dots* and *squares* indicate the training and test samples

the SDPP and its *kernelized* version have demonstrated the effectiveness of the proposed approach for supervised DR. The proposed method has been capable of retrieving meaningful low-dimensional subspaces that are useful for data exploration and suitable for learning accurate regression and classification models. Empirically, the SDPP and the KSDPP performed comparably or superiorly to the other techniques considered in this study, thus making it a simple, computationally light and yet accurate tool for data analysis.

Appendix A: Derivation of a Compact form for $\nabla_{\mathbf{W}} J$

We want to rewrite the gradient $\nabla_{\mathbf{W}} J = \frac{4}{n} \sum_{ij} \mathbf{G}_{ij} (\mathbf{D}_{ij} - \boldsymbol{\Delta}_{ij}) \boldsymbol{\tau}_{ij} \boldsymbol{\tau}_{ij}^T \mathbf{W}$ into a more compact form. Firstly, we denote $\mathbf{Q} = \mathbf{G} \odot (\mathbf{D} - \boldsymbol{\Delta})$, where $\odot$ represents the element-wise product of two matrices, symmetric matrix $\mathbf{R} = \mathbf{Q} + \mathbf{Q}^T$, and $\mathbf{S}$ is a diagonal matrix with $\mathbf{S}_{ii} = \sum_j \mathbf{R}_{ij}$. Then, we manipulate $\nabla_{\mathbf{W}} J$ as follows:

$$\begin{aligned} \nabla_{\mathbf{W}} J &= \frac{4}{n} \sum_{ij} \mathbf{Q}_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W} \\ &= \frac{4}{n} \sum_{ij} (\mathbf{x}_i \mathbf{Q}_{ij} \mathbf{x}_i^T + \mathbf{x}_j \mathbf{Q}_{ij} \mathbf{x}_j^T - \mathbf{x}_i \mathbf{Q}_{ij} \mathbf{x}_j^T - \mathbf{x}_j \mathbf{Q}_{ij} \mathbf{x}_i^T) \mathbf{W} \end{aligned}$$

$$
\begin{aligned}
&= \frac{4}{n}\left[\sum_{ij}\mathbf{x}_i(\mathbf{Q}_{ij}+\mathbf{Q}_{ji})\mathbf{x}_i^T - \sum_{ij}\mathbf{x}_i(\mathbf{Q}_{ij}+\mathbf{Q}_{ji})\mathbf{x}_j^T\right]\mathbf{W}\\
&= \frac{4}{n}\left(\sum_{ij}\mathbf{x}_i\mathbf{R}_{ij}\mathbf{x}_i^T - \sum_{ij}\mathbf{x}_i\mathbf{R}_{ij}\mathbf{x}_j^T\right)\mathbf{W}\\
&= \frac{4}{n}\left(\sum_{i}\mathbf{x}_i\sum_{j}\mathbf{R}_{ij}\mathbf{x}_j^T - \sum_{ij}\mathbf{x}_i\mathbf{R}_{ij}\mathbf{x}_j^T\right)\mathbf{W}\\
&= \frac{4}{n}\left(\sum_{i}\mathbf{x}_i\mathbf{S}_{ii}\mathbf{x}_j^T - \sum_{ij}\mathbf{x}_i\mathbf{R}_{ij}\mathbf{x}_j^T\right)\mathbf{W}\\
&= \frac{4}{n}\left(\mathbf{X}^T\mathbf{S}\mathbf{X} - \mathbf{X}^T\mathbf{R}\mathbf{X}\right)\mathbf{W}\\
&= \frac{4}{n}\mathbf{X}^T(\mathbf{S}-\mathbf{R})\mathbf{X}\mathbf{W},
\end{aligned}
$$

where each row of data matrix $\mathbf{X}$ is a data point $\mathbf{x}_i$ and $\mathbf{L} = \mathbf{S} - \mathbf{R}$ is the Laplacian matrix.

References

1. Asuncion A, Newman D (2007) UCI machine learning repository. Report, University of California, Irvine
2. Barshan E, Ghodsi A, Azimifar Z, Zolghadri Jahromi M (2010) Zolghadri Jahromi, M.: Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recognit 44:1357–1371
3. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
4. Belkin M, Niyogi P (2002) Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in neural information processing systems, Cambridge, pp 585–592
5. Fukumizu K, Bach FR, Jordan M (2009) Kernel dimension reduction in regression. Ann Stat 37:1871–1905
6. Globerson A, Roweis S (2005) Metric learning by collapsing classes. In. Advances in neural information processing systems, Vancouver, pp 451–458
7. Gretton A, Bousquet O, Smola A, Schölkopf B (2005) Measuring statistical dependence with Hilbert–Schmidt norms. In: Algorithmic learning theory, Singapore, pp 63–77
8. Groenen P, van de Velden M (2004) Multidimensional scaling. Technical Report EI 2004–15, Erasmus University, Rotterdam
9. Hardoon D, Szedmak S, Shawe-Taylor J (2004) Canonical correlation analysis: an overview with application to learning methods. Neural Comput 16(12):2639–2664
10. Hofmann T, Schölkopf B, Smola A (2008) Kernel methods in machine learning. Ann Stat 36(3):1171–1220
11. Lee J, Verleysen M (2007) Nonlinear dimensionality reduction. Springer, New York
12. Li K (1991) Sliced inverse regression for dimension reduction. J Am Stat Assoc 86(414):316–327
13. Li L, Liu J (2009) Constrained clustering by spectral kernel learning. In: International conference on computer vision, Kyoto, pp 421–427
14. Mika S, Ratsch G, Weston J, Scholkopf B, Mullers K (1999) Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop. IEEE, Piscataway, pp 41–48
15. Rosipal R, Krämer N (2006) Overview and recent advances in partial least squares. In: Subspace, latent structure and feature selection, Bohinj, pp 34–51
16. Rosipal R, Trejo L (2002) Kernel partial least squares regression in reproducing kernel Hilbert space. J Mach Learn Res 2:97–123

17. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
18. Schölkopf B, Smola A, Müller K (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural comput 10(5):1299–1319
19. Sha F, Saul L (2005) Analysis and extension of spectral methods for nonlinear dimensionality reduction. In: Proceedings of the twenty second international conference on, machine learning, Bonn, pp 785–792
20. Takane Y, Young F, De Leeuw J (1977) Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. Psychometrika 42(1):7–67
21. Tenenbaum J, Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
22. Tipping M, Lowe D (1998) Shadow targets: a novel algorithm for topographic projections by radial basis functions. Neurocomputing 19:211–222
23. Toh K, Todd M, Tutuncu R (1999) Sdpt3a matlab software package for semidefinite programming. Optim Methods Softw 11(12):545–581
24. Tütüncü R, Toh K, Todd M (2003) Solving semidefinite-quadratic-linear programs using SDPT3. Math Program 95(2):189–217
25. van der Maaten L, Postma E, van den Herik H (2009) Dimensionality reduction: a comparative review. Technical Report TiCC-TR 2009–005, Tilburg University Technical, Tilburg
26. Venna J, Kaski S (2007) Comparison of visualization methods for an atlas of gene expression data sets. Inf Vis 6:139–154
27. Webb A (1995) Multidimensional scaling by iterative majorization using radial basis functions. Pattern Recognit 28:753–759
28. Weinberger K, Sha F, Saul L (2004) Learning a kernel matrix for nonlinear dimensionality reduction. In: Proceedings of the 21st international conference on, machine learning, Banff
29. Weinberger K, Blitzer J, Saul L (2006) Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems, Vancouver
30. Weinberger K, Sha F, Zhu Q, Saul L (2006) Graph laplacian regularization for large-scale semidefinite programming. In: Advances in neural information processing systems, Vancouver, pp 1489–1496
31. Wold H (1975) Soft modeling by latent variables: the nonlinear iterative partial least squares approach. Perspectives in probability and statistics, papers in honour of MS Bartlett, pp 520–540
32. Wu X, So A, Li Z, Li S (2009) Fast graph Laplacian regularized kernel learning via semidefinite-quadratic-linear programming. In: Advances in neural information processing systems, Vancouver, pp 1964–1972 (2009)
33. Xing E, Ng A, Jordan M, Russell S (2003) Distance metric learning with application to clustering with side-information. In: Advances in neural information processing systems, Vancouver, pp 521–528
34. Yeh Y, Huang S, Lee Y (2009) Nonlinear dimension reduction with kernel sliced inverse regression. IEEE Trans Knowl Data Eng 21:1590–1603