Computing and Informatics, Vol. 38, 2019, 917–936, doi: 10.31577/cai_2019_4_917

HSIC REGULARIZED LTSA

Xinghua Zheng

School of Data and Computer Science, Sun Yat-sen University
Guangzhou, 510275, China
e-mail: 380567579@qq.com

Zhengming Ma*, Hangjian Che

School of Electronics and Information Technology, Sun Yat-sen University
Guangzhou, 510275, China
e-mail: issmzm@mail.sysu.edu.cn, 1349831461@qq.com

Lei Li

School of Data and Computer Science, Sun Yat-sen University
Guangzhou, 510275, China
e-mail: leili525@126.com

Abstract. Hilbert-Schmidt Independence Criterion (HSIC) measures statistical independence between two random variables. However, instead of measuring the statistical independence between two random variables directly, HSIC first transforms two random variables into two Reproducing Kernel Hilbert Spaces (RKHS) respectively and then measures the kernelled random variables by using Hilbert-Schmidt (HS) operators between the two RKHS. Since HSIC was first proposed around 2005, HSIC has found wide applications in machine learning. In this paper, a HSIC regularized Local Tangent Space Alignment algorithm (HSIC-LTSA) is proposed. LTSA is a well-known dimensionality reduction algorithm for local homeomorphism preservation. In HSIC-LTSA, behind the objective function of LTSA, HSIC between high-dimensional and dimension-reduced data is added as a regularization term. The proposed HSIC-LTSA has two contributions. First,

[*] Corresponding author

HSIC-LTSA implements local homeomorphism preservation and global statistical correlation during dimensionality reduction. Secondly, HSIC-LTSA proposes a new way to apply HSIC: HSIC is used as a regularization term to be added to other machine learning algorithms. The experimental results presented in this paper show that HSIC-LTSA can achieve better performance than the original LTSA.

Keywords: Dimensionality reduction, RKHS, Hilbert-Schmidt operators, LTSA, HSIC

1 INTRODUCTION

The loss of information is inevitable during dimensionality reduction. Therefore, the main concern in constructing algorithms of dimensionality reduction is what information needs to be preserved during dimensionality reduction. From this viewpoint, the algorithms of dimensionality reduction can be divided into two categories: global-preserving and local-preserving algorithms [1]. The global-preserving algorithms preserve some global features of data during dimensionality reduction [2, 3, 4, 5], while the local-preserving algorithms preserve some local features of data during dimensionality reduction. Local Tangent Space Alignment (LTSA) algorithm is a typical local-preserving algorithm for dimensionality reduction. The local feature LTSA preserves is the local homeomorphism, i.e., the continuous dependence between data within a local region [6]. In recent years, the dimensionality reduction algorithms capable of preserving both local and global features have emerged, such as LPP [7, 8, 9].

Hilbert-Schmidt Independence Criterion (HSIC) measures the statistical independence between two random variables [10]. However, instead of measuring the statistical independence between two random variables directly, HSIC first transforms the two random variables into two reproducing kernel Hilbert spaces (RKHS) respectively and then measures the statistical correlation of the two transformed random variables by using Hilbert-Schmidt operators between two RKHSs. In the application of HSIC to data analysis, the given data can be regarded as the values taken by the random variables. The HSIC formulae for calculating the statistical correlation of data are simple and often used in many applications [11, 12, 13, 14, 15]. However, HSIC involves many mathematical concepts and it is not easy to understand the meaning of HSIC thoroughly. The misunderstanding, or even misuse of HSIC happens from time to time.

In this paper, HSIC is first explored theoretically and then applied to LTSA. LTSA is a local homeomorphism-preserving algorithm for dimensionality reduction. An improved LTSA, called HSIC regularized LTSA, or HSIC-LTSA for short, is proposed in which a HSIC regularization term is added to LTSA's objective function. The HSIC regularization term measures the statistical correlation between the high-dimensional data and the dimension-reduced data. HSIC-LTSA takes into account both the local and global preserving requirements during dimensionality reduction and achieves a better result than LTSA.

The remaining sections in this paper are arranged as follows: in Section 2, LTSA is elaborated, showing that LTSA is a local-homeomorphism preserving algorithm in nature; in Section 3, RKHS is briefly described; in Section 4, the theory of HSIC is elaborated thoroughly and the HSIC formulae for calculating the statistical correlation between two sets of data is derived. In Section 5, an improved HSIC-LTSA is proposed; in Section 6, the experimental results of LTSA and HSIC-LTSA are presented to show the effectiveness of HSIC-LTSA; in Section 7, some conclusions are presented.

2 LOCAL TANGENT SPACE ALIGNMENT (LTSA)

LTSA [6] is a classical local homeomorphism-preserving algorithm of manifold learning and mainly applied to dimensionality reduction. Generally speaking, the problem of dimensionality reduction can be expressed as follows: given a set of high-dimensional data $X = \{x_1, \ldots, x_N\} \subseteq R^D$, we want to find an another set of data $Y = \{y_1, \ldots, y_N\} \subseteq R^d$ such that y_n is the dimensional-reduced version of x_n, where $d << D$ and $n = 1, \ldots, N$. In manifold learning, Y is also called the global coordinate of X.

Remark 1. In this paper, a dataset can be represented by a set, in which the elements of the set are data, for example, $X = \{x_1, \ldots, x_N\} \subseteq R^D$. The dataset can also be represented by a matrix, in which the column vectors of matrix are data, for example, $X = [x_1, \ldots, x_N] \in R^{D\times N}$. The two representations are equivalent.

The stages of LTSA are as follows:

1. Decompose the high-dimensional data into local groups: LTSA uses K-NN method. For each data x_n, let $x_{n_1}, \ldots, x_{n_K}$ be its K-nearest neighbors, then the n^{th} local group is as follows:

$$X_n = \left[x_{n_1} \ldots x_{n_K}, x_{n_{K+1}}\right] \in R^{D\times(K+1)} \tag{1}$$

where $x_{n_{K+1}} = x_n$, $n = 1, \ldots, N$. It is clear that $X = \bigcup_{n=1}^{N} X_n$.

2. Reduce the dimension of each local group X_n: LTSA uses PCA method. The local group X_n is first centralized:

$$\hat{X}_n = \left[x_{n_1} - \bar{x}_n, x_{n_{K+1}} - \bar{x}_n\right] = X_n C_{K+1} \tag{2}$$

where $\bar{x}_n = \frac{1}{K+1}\sum_{k=1}^{K+1} x_{n_k}$, $C_{K+1} = I_{K+1} - \frac{1}{K+1}\Gamma_{K+1}\Gamma_{K+1}^T$, $\Gamma_{K+1} = \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} \in R^{K+1}$. C_{K+1} is often called centralizing matrix.

Then, the centralized matrix $\hat{X}_n$ is SVD-decomposed:

$$\hat{X}_n = U_n \Sigma_n V_n^T \tag{3}$$

where both $U_n \in R^{D\times D}$ and $V_n \in R^{(K+1)\times(K+1)}$ are orthonormal matrices, $\Sigma_n \in R^{D\times(K+1)(K+1)}$ is singular value matrix.

At last, let $U_{n,1} \in R^{D\times d}$ be the matrix composed of the first d column vectors of U_n, then the column vectors of $\hat{X}_n$ are projected into the space spanned by the column vectors of $U_{n,1}$, i.e., $spanU_{n,1}$, the coordinates of these projections in $spanU_{n,1}$ are the dimensional-reduced version of X_n:

$$\Theta_n = U_{n,1}^T \hat{X}_n \in R^{d\times(K+1)}. \tag{4}$$

In manifold learning, Θ_n is often called the local coordinate of X_n.

Remark 2. From the viewpoint of manifold, the space $spanU_{n,1}$ can be regarded as the tangent space [16] of the point $\bar{x}_n$, therefore LTSA is called local tangent space method. Furthermore, since $\hat{X}_n$ and Θ_n are homeomorphic [17] to each other within the neighborhood of $\bar{x}_n$, therefore, LTSA belongs to the category of local preserving algorithms.

3. Derive the global coordinate from the local coordinate: Let us denote the global coordinate of X_n as

$$Y_n = \left[y_{n_1} \dots y_{n_{K+1}}\right] \in R^{d\times(K+1)} \tag{5}$$

where y_{n_k} is the global coordinate of x_{n_k}, i.e., the dimensional-reduced version of x_{n_k}, $1 \le n_k \le N$, $k = 1, \dots, K+1$. We want to derive Y_n from Θ_n. LTSA assumes that Y_n is the linear transformation of Θ_n (affine transformation, strictly speaking):

$$\hat{Y}_n = \left[y_{n_1} - \bar{y}_n \dots y_{n_{K+1}} - \bar{y}_n\right] = Y_n C_{K+1} = A_n \Theta_n \tag{6}$$

where $\bar{y}_n = \frac{1}{K+1}\sum_{k=1}^{K+1} y_{n_k}$. The geometric meaning of Equation (6) is that Y_n can be derived from Θ_n by translation, rotation and scale. Furthermore,

$$\hat{Y}_n = A_n \Theta_n \Rightarrow A_n = \hat{Y}_n \Theta_n^+ \tag{7}$$

where Θ_n^+ represents the right pseudo inverse of Θ_n, i.e., Θ_n^+ is the solution to the following problem:

$$\left\| I_d - \Theta_n \Theta_n^+ \right\|^2 \xrightarrow[choose\ \Theta_n^+]{} \min. \tag{8}$$

Based on Equation (7), the local objective function can be established:

$$\left\| \hat{Y}_n - A_n \Theta_n \right\|^2 = \left\| \hat{Y}_n \left(I_{K+1} - \Theta_n^+ \Theta_n\right) \right\|^2 = \left\| Y_n C_{K+1} \left(I_{K+1} - \Theta_n^+ \Theta_n\right) \right\|^2$$

$$= \left\| Y S_n C_{K+1} \left(I_{K+1} - \Theta_n^+ \Theta_n \right) \right\|^2 = \left\| Y L_n \right\|^2 \xrightarrow[choose\ Y]{} \min \tag{9}$$

where $S_n \in R^{N \times (K+1)}$ is the selection matrix such that $Y_n = Y S_n$, i.e., the n_k^{th} element of the k^{th} column vector is 1, other elements are 0, $k = 1, \ldots, K+1$; $L_n = S_n C_{K+1} \left(I_{K+1} - \Theta_n^+ \Theta_n \right)$, called the local pattern of X.

The objective function of LTSA can be derived by summing up all the local objective functions:

$$\sum_{n=1}^{N} \left\| Y L_n \right\|^2 = \sum_{n=1}^{N} tr \left(Y L_n L_n^T Y^T \right) = tr \left(Y \sum_{n=1}^{N} L_n L_n^T Y^T \right)$$
$$= tr \left(Y L L^T Y^T \right) \xrightarrow[choose\ Y]{} \min \tag{10}$$

where $L = [L_1 \ldots L_N]$.

3 REPRODUCING KERNEL HILBERT SPACES (RKHS)

HSIC is based on RKHS. Let $L^2(\Omega) = \left\{ f \left| f : \Omega \to R, \int_\Omega |f(x)|^2 < +\infty \right. \right\}$ be the space of square integrable functions. An inner product $\langle \bullet, \bullet \rangle$ can be defined over $L^2(\Omega)$ [18]:

$$\langle f, g \rangle = \int_\Omega f(x) g(x) dx. \tag{11}$$

It can be proven that $H = \left(L^2(\Omega), \langle \bullet, \bullet \rangle \right)$ is a complete inner product space, i.e., Hilbert space.

Definition 1 ([18])**.** Let $H = \left(L^2(\Omega), \langle \bullet, \bullet \rangle \right)$, if there is a function $k : \Omega \times \Omega \to R$ such that

- for all $x \in \Omega$, $k_x = k(\bullet, x) \in H$,
- for all $f \in H$, $f(x) = \langle f, k(\bullet, x) \rangle$,

then H is called a reproducing kernel Hilbert space (RKHS) and k is called the reproducing kernel of H.

The reproducing kernel k can be used to define a map: $\varphi : \Omega \to H$ such that for all $x \in \Omega$,

$$\varphi(x) = k(\bullet, x) \in H. \tag{12}$$

It can be easily proven that

$$\langle \varphi(x), \varphi(y) \rangle = \langle k_x, k(\bullet, y) \rangle = k_x(y) = k(y, x) = k(x, y). \tag{13}$$

The above equation is often used in many kernel methods of machine learning such as kPCA [3], kLDA [19], kSVM [20], etc.

Furthermore, if X is a random variable on Ω, then $\varphi(X)$ is a random process and its mean function is defined:

$$\mu_X(u) = E_X[\varphi(X)(u)] = E_X[k(u,X)] = \int_\Omega k(u,x)\,p_X(x)\,\mathrm{d}x. \tag{14}$$

Then, for all $f \in H$,

$$\begin{aligned}\langle \mu_X, f\rangle &= \int_\Omega \mu_X(u)\,f(u)\,\mathrm{d}u = \int_\Omega \left(\int_\Omega k(u,x)\,p_X(x)\,\mathrm{d}x\right) f(u)\,\mathrm{d}u \\ &= \int_\Omega \left(\int_\Omega k(u,x)\,f(u)\,\mathrm{d}u\right) p_X(x)\,\mathrm{d}x = \int_\Omega \langle f, k(\bullet,x)\rangle\, p_X(x)\,\mathrm{d}x \\ &= \int_\Omega f(x)\,p_x(x)\,\mathrm{d}x = E_x[f(X)]. \end{aligned} \tag{15}$$

In mathematics, it can be proven that RKHS can be generated from kernel functions. The definition of kernel functions is as follows.

Definition 2 ([21])**.** Let $k : \Omega \times \Omega \to R$, if k satisfies the following conditions:

- Symmetric: for all $x, y \in \Omega$, $k(x,y) = k(y,x)$,
- Square integrable: for all $x \in \Omega$, $k_x = k(\bullet, x)$ is square integrable,
- Positive definite: for all $x_1, \ldots, x_N \in \Omega$, the matrix $\begin{bmatrix} k(x_1,x_1) & \ldots & k(x_1,x_N) \\ \vdots & \ddots & \vdots \\ k(x_N,x_1) & \ldots & k(x_N,x_N) \end{bmatrix}$ is positive definite,

then k is called a kernel function.

Remark 3. Kernel functions and reproducing kernels are not the same concept. Kernel functions are defined on their own, while reproducing kernels are defined based on RKHS.

Theorem 1 ([18])**.** A kernel function k can be used to generate a unique RHHS H_k such that k becomes the reproducing kernel of H_k.

Based on this theorem, as long as we define a kernel function, we define an RKHS.

4 HILBERT-SCHMIDT INDEPENDENCE CRITERION (HSIC)

4.1 HS Operators

HSIC is defined by using Hilbert-Schmidt operators.

Definition 3 ([22])**.** Let H_X and H_Y be two separable Hilbert spaces, $\{e_i \,|i \in I\}$ the orthonormal basis of H_X, $T : H_X \to H_Y$ a compact operator, if $\sum_{i \in I} \|Te_i\|_Y^2 < +\infty$, then T is called a Hilbert-Schmidt (HS) operator.

Remark 4. In this paper, $\langle \bullet, \bullet \rangle_X$ represents the inner product of H_X, $\|\bullet\|_X = \sqrt{\langle \bullet, \bullet \rangle_X}$ the norm of H_X. Similarly, $\langle \bullet, \bullet \rangle_Y$ represents the inner product of H_Y, $\|\bullet\|_Y = \sqrt{\langle \bullet, \bullet \rangle_Y}$ the norm of H_Y.

Let $HS\left(H_X \to H_Y\right)$ be the space of all HS operators from H_X to H_Y. An inner product $\langle \bullet, \bullet \rangle_{HS}$ can be defined on $HS\left(H_X \to H_Y\right)$ to make $(HS(H_X \to H_Y)$, $\langle \bullet, \bullet \rangle_{HS})$ become a Hilbert space.

Theorem 2 ([10])**.** If for all $T, S \in HS\left(H_X \to H_Y\right)$, $\sum_{i \in I} |\langle Te_i, Se_i \rangle_Y| < +\infty$, then $\left(HS\left(H_X \to H_Y\right), \langle \bullet, \bullet \rangle_{HS}\right)$ is a Hilbert space, where the inner product $\langle \bullet, \bullet \rangle_{HS}$ is defined as follows:

$$\langle T, S \rangle_{HS} = \sum_{i \in I} \langle Te_i, Se_i \rangle_Y. \tag{16}$$

Tensor product operators are a kind of HS operators.

Theorem 3 ([10])**.** Let H_X and H_Y be two separable Hilbert spaces, $f_0 \in H_X$, $g_0 \in H_X$, define $f_0 \otimes g_0 : H_X \to H_Y$ such that for all $f \in H_X$, $f_0 \otimes g_0\,(f) = \langle f_0, f \rangle_X g_0 \in H_Y$, then $f_0 \otimes g_0$ is a HS operator, i.e., $f_0 \otimes g_0 \in HS\left(H_X \to H_Y\right)$.

Remark 5. $f_0 \otimes g_0$ is called the tensor product of f_0 and g_0.

4.2 Cross Covariance Operators

Generally speaking, HSIC involves two RKHSs.

Let $H_1 = \left(L^2\left(\Omega_1\right), \langle \bullet, \bullet \rangle_1\right)$ be an RKHS, $k_1 : \Omega_1 \times \Omega_1 \to R$ the reproducing kernel of H_1. Define $\varphi_1 : \Omega_1 \to H_1$ such that for all $x \in \Omega_1$, $\varphi_1\left(x\right) = k_1\left(\bullet, x\right) \in H_1$. Note that $\langle \varphi_1\left(x'\right), \varphi_1\left(x''\right) \rangle_1 = k_1\left(x', x''\right)$.

Similarly, let $H_2 = \left(L^2\left(\Omega_2\right), \langle \bullet, \bullet \rangle_2\right)$ be an RKHS, $k_2 : \Omega_2 \times \Omega_2 \to R$ the reproducing kernel of H_2. Define $\varphi_2 : \Omega_2 \to H_2$ such that for all $y \in \Omega_2$, $\varphi_2\left(y\right) = k_2\left(\bullet, y\right) \in H_2$. Note that $\langle \varphi_2\left(y'\right), \varphi_2\left(y''\right) \rangle_2 = k_2\left(y', y''\right)$.

Furthermore, let X be a random variable on Ω_1, Y a random variable on Ω_2.

Theorem 4 ([10])**.** Let $\Phi : HS\left(H_1 \to H_2\right) \to R$ such that for all $T \in HS(H_1 \to H_2)$

$$\Phi\left(T\right) = E_{XY}\left[\langle \varphi_1\left(X\right) \otimes \varphi_2\left(Y\right), T \rangle_{HS}\right]. \tag{17}$$

If $E_{XY}\left[\|\varphi_1\left(X\right) \otimes \varphi_2\left(Y\right)\|_{HS}\right] < +\infty$, then Φ is continuous linear functional on $HS\left(H_1 \to H_2\right)$.

According to the representation theorem of continuous linear functionals (Riesz theorem [18]), there must be a unique HS operator $T_\Phi \in HS\left(H_X \to H_Y\right)$ such that for all HS operators $T \in HS\left(H_X \to H_Y\right)$,

$$\Phi\left(T\right) = E_{XY}\left[\langle \varphi_1\left(X\right) \otimes \varphi_2\left(Y\right), T \rangle_{HS}\right] = \langle T, T_\Phi \rangle_{HS}. \tag{18}$$

This HS operator T_Φ is called as cross covariance operator and often denoted as C_{XY}.

4.3 Hilbert-Schmidt Independence Criterion (HSIC)

Definition 4 ([10])**.** The HSIC of two random variables X and Y is defined as

$$HSIC\left(X,Y\right) = E_{XY}\left[\|(\varphi_1\left(X\right) - \mu_X) \otimes (\varphi_2\left(Y\right) - \mu_Y)\|_{HS}^2\right]. \tag{19}$$

It can be easily proven [10] that:

$$\begin{aligned} HSIC\left(X,Y\right) &= E_{XY}\left[\|(\varphi_1\left(X\right) - \mu_X) \otimes (\varphi_2\left(Y\right) - \mu_Y)\|_{HS}^2\right] \\ &= \|C_{XY} - \mu_X \otimes \mu_Y\|_{HS}^2 \\ &= \langle C_{XY},C_{XY}\rangle_{HS} - 2\langle C_{XY}, \mu_X \otimes \mu_Y\rangle_{HS} \\ &\quad + \langle \mu_X \otimes \mu_Y, \mu_X \otimes \mu_Y\rangle_{HS}. \end{aligned} \tag{20}$$

In practice, two sets of data $\{x_1, \ldots, x_N\} \subseteq \Omega_1$ and $\{y_1, \ldots, y_N\} \subseteq \Omega_2$ are given and can be regarded as some sample taken by the random variables X and Y. Therefore, the calculation of HSIC can be approximated by replacing statistical average with sample average [10].

At first, for all HS operators $T \in HS\left(H_1 \to H_2\right)$, since

$$\begin{aligned} \langle C_{XY}, T\rangle_{HS} &= E_{XY}\left[\langle \varphi_1\left(X\right) \otimes \varphi_2\left(Y\right), T\rangle_{HS}\right] \\ &\approx \frac{1}{N}\sum_{n=1}^{N} \langle \varphi_1\left(x_n\right) \otimes \varphi_2\left(y_n\right), T\rangle_{HS} \\ &= \left\langle \frac{1}{N}\sum_{n=1}^{N} \varphi_1\left(x_n\right) \otimes \varphi_2\left(y_n\right), T\right\rangle_{HS} \end{aligned} \tag{21}$$

then

$$C_{XY} \approx \frac{1}{N}\sum_{n=1}^{N} \varphi_1\left(x_n\right) \otimes \varphi_2\left(y_n\right). \tag{22}$$

Similarly, for all functions $f \in H_1$, since

$$\langle f, \mu_X\rangle_1 = E_X\left[\langle \varphi_1\left(X\right), f\rangle_1\right] \approx \frac{1}{N}\sum_{n=1}^{N} \langle \varphi_1\left(x_n\right), f\rangle_1 = \left\langle \frac{1}{N}\sum_{n=1}^{N} \varphi_1\left(x_n\right), f\right\rangle_1$$

then

$$\mu_X \approx \frac{1}{N}\sum_{n=1}^{N} \varphi_1\left(x_n\right). \tag{23}$$

By the same deduction, we have

$$\mu_Y \approx \frac{1}{N}\sum_{n=1}^{N}\varphi_2\left(y_n\right). \tag{24}$$

Substituting Equations (34), (35), (36) into Equations (31), (32), (33) gives:

$$\begin{aligned}\langle C_{XY}, C_{XY}\rangle_{HS} &\approx \left\langle \frac{1}{N}\sum_{i=1}^{N}\varphi_1\left(x_i\right)\otimes\varphi_2\left(y_i\right), \frac{1}{N}\sum_{j=1}^{N}\varphi_1\left(x_j\right)\otimes\varphi_2\left(y_j\right)\right\rangle_{HS}\\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}k_1\left(x_i,x_j\right)k_2\left(y_i,y_j\right) = \frac{1}{N^2}tr\left(K_1K_2\right), \qquad (25)\end{aligned}$$

$$\begin{aligned}\langle C_{XY}, \mu_X\otimes\mu_Y\rangle_{HS} &\approx \left\langle \frac{1}{N}\sum_{i=1}^{N}\varphi_1\left(x_i\right)\otimes\varphi_2\left(y_i\right),\right.\\ &\quad \left.\left(\frac{1}{N}\sum_{p=1}^{N}\varphi_1\left(x_p\right)\right)\otimes\left(\frac{1}{N}\sum_{q=1}^{N}\varphi_2\left(y_q\right)\right)\right\rangle_{HS}\\ &= \frac{1}{N^3}\sum_{i=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N}k_1\left(x_i,x_p\right)k_2\left(y_i,y_q\right) = \frac{1}{N^3}\Gamma_N^TK_1K_2\Gamma_N \qquad (26)\\ &\quad \langle\mu_X\otimes\mu_Y, \mu_X\otimes\mu_Y\rangle_{HS} = \langle\mu_X,\mu_X\rangle_1\langle\mu_Y,\mu_Y\rangle_2\\ &\approx \left\langle\frac{1}{N}\sum_{i=1}^{N}\varphi_1(x_i), \frac{1}{N}\sum_{j=1}^{N}\varphi_1(x_j)\right\rangle_1\left\langle\frac{1}{N}\sum_{i=1}^{N}\varphi_2(y_i), \frac{1}{N}\sum_{j=1}^{N}\varphi_2(y_j)\right\rangle_2\\ &= \frac{1}{N^4}\Gamma_N^TK_1\Gamma_N\Gamma_N^TK_2\Gamma_N \qquad (27)\end{aligned}$$

where

$$K_1 = \begin{bmatrix} k_1\left(x_1,x_1\right) & \dots & k_1\left(x_1,x_N\right)\\ \vdots & \ddots & \vdots\\ k_1\left(x_N,x_1\right) & \dots & k_1\left(x_N,x_N\right)\end{bmatrix}, \quad K_2 = \begin{bmatrix} k_2\left(y_1,y_1\right) & \dots & k_2\left(y_1,y_N\right)\\ \vdots & \ddots & \vdots\\ k_2\left(y_N,y_1\right) & \dots & k_2\left(y_N,y_N\right)\end{bmatrix}. \tag{28}$$

Substituting (37), (38), (39) into Equation (30) gives:

$$\begin{aligned} HSIC\left(X,Y\right) &= \langle C_{XY}, C_{XY}\rangle_{HS} - 2\langle C_{XY},\mu_X\otimes\mu_Y\rangle_{HS} + \langle\mu_X\otimes\mu_Y,\mu_X\otimes\mu_Y\rangle_{HS}\\ &\approx \frac{1}{N^2}tr\left(K_1K_2\right) - \frac{2}{N^3}\Gamma_N^TK_1K_2\Gamma_N + \frac{1}{N^4}\Gamma_N^TK_1\Gamma_N\Gamma_N^TK_2\Gamma_N \qquad (29)\\ &= \frac{1}{N^2}tr\left(K_2C_NK_1C_N\right) \qquad (30)\end{aligned}$$

where $C_N = I_N - \frac{1}{N}\Gamma_N \Gamma_N^T$ is the centralized matrix.

5 HSIC REGULARIZED LTSA (HSIC-LTSA)

5.1 The Objective Function of HSIC-LTSA

In manifold learning, LTSA is among the few algorithms which are created based on the mathematical properties of manifolds. Therefore, LTSA achieves better performance in the process of manifold data. However, the so-called manifolds are topological spaces which are locally homeomorphic to Euclidean spaces. Therefore, it is natural for LTSA to be a local homeomorphism-preserving algorithm. Many improvements to LTSA try to turn LTSA into one capable of preserving both local and global properties of data during dimensionality reduction. For example, in [23], the dimension-reduced data Y are set to the linear transformation of the high dimensional data X, i.e., $Y = WX$, where $W \in R^{d\times D}$. Y is then replaced with $Y = WX$ in the objective function of LTSA:

$$tr\left(YLL^TY^T\right) \xrightarrow[\text{choose } Y]{} \min \Rightarrow tr\left(WXLL^TX^TW^T\right) \xrightarrow[\text{choose } W]{} \min \tag{31}$$

However, the setting $Y = WX$ will destroy the nonlinear nature of LTSA.

In this paper, an improved LTSA, called HSIC regularized LTSA (HSIC-LTSA for short), is proposed in which a HSIC regularization term is added to the objective function of LTSA:

$$tr\left(YLL^TY^T\right) - \lambda HSIC\left(X,Y\right) = tr\left(YLL^TY^T\right) - \lambda tr\left(K_2C_NK_1C_N\right) \xrightarrow[\text{choose } Y]{} \min \tag{32}$$

where $\lambda > 0$ is the regularization coefficient.

$HSIC\left(X,Y\right)$ measures the statistical dependence of two random processes $\varphi_1\left(X\right)$ and $\varphi_2\left(Y\right)$. Therefore, the objective function HSIC-LTSA shown in Equation (32) means that X and Y will be kept statistically dependent as much as possible during dimensionality reduction of LTSA.

Furthermore, the dimension-reduced data Y is hidden in the kernel matrix K_2 in $HSIC\left(X,Y\right)$. In order to facilitate the optimization of Y, the proposed HSIC-LTSA sets the kernel function k_2 based on the linear kernel: $k_2 : R^d \times R^d \to R$, for all $y', y'' \in R^d$,

$$k_2\left(y', y''\right) = {y'}^T y'' + \kappa\delta\left(y', y''\right) \tag{33}$$

where $\kappa > 0$ and $\delta\left(y', y''\right) = \begin{cases} 1, & y' = y'' \\ 0, & \text{others} \end{cases}$. The addition of δ ensures the positive definiteness of k_2. The kernel matrix K_2 is then to be:

$$K_2 = \begin{bmatrix} k_2(y_1, y_1) & \dots & k_2(y_1, y_N) \\ \vdots & \ddots & \vdots \\ k_2(y_N, y_1) & \dots & k_2(y_N, y_N) \end{bmatrix} = \begin{bmatrix} y_1^T y_1 & \dots & y_1^T y_N \\ \vdots & \ddots & \vdots \\ y_N^T y_1 & \dots & y_N^T y_N \end{bmatrix} + \kappa I_N = Y^T Y + \kappa I_N. \tag{34}$$

In this setting of K_2, $HSIC(X, Y)$ will become:

$$\begin{aligned} HSIC(X, Y) &= tr(K_2 C_N K_1 C_N) = tr\left(Y^T Y C_N K_1 C_N\right) + \kappa tr(C_N K_1 C_N) \\ &= tr\left(Y C_N K_1 C_N Y^T\right) + \kappa tr(C_N K_1 C_N). \end{aligned} \tag{35}$$

$tr(C_N K_1 C_N)$ has nothing to do with Y, therefore the objective function of HSIC-LTSA becomes:

$$tr\left(Y L L^T Y^T\right) - \lambda tr\left(Y C_N K_1 C_N Y^T\right) \xrightarrow[\text{choose } Y]{} \min \tag{36}$$

where

$$K_1 = \begin{bmatrix} k_1(x_1, x_1) & \dots & k_1(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k_1(x_N, x_1) & \dots & k_1(x_N, x_N) \end{bmatrix}. \tag{37}$$

The kernel function k_1 can be chosen according to the applications at hand. Therefore, HSIC-LTSA provides much flexibility for different applications.

5.2 The Solution to HSIC Regularized LTSA

The objective function of HSIC-LTSA shown in Equation (37) can be rewritten in an equivalent form:

$$\frac{tr\left(Y L L^T Y^T\right)}{tr\left(Y C_N K_1 C_N Y^T\right)} \xrightarrow[\text{choose } Y]{} \min. \tag{38}$$

In Equation (38), since for all constant vectors $z \in R^N$, $C_N z = 0$, $C_N K_1 C_N$ is then positive semi-definite, not positive definite. However, from another viewpoint, C_N is the centralizing matrix, $Y C_N$ means the centralization of Y. In geometry, $Y C_N$ means translation of Y to the origin of the Euclidean space R^n. Obviously, the translation of Y has no impact on the result of dimensionality reduction. Therefore, it is reasonable to assume that $Y C_N = Y$. Under this assumption, the objective function shown in Equation (38) can be refined as follows:

$$\frac{tr\left(Y L L^T Y^T\right)}{tr\left(Y K_1 Y^T\right)} \xrightarrow[\text{choose } Y]{} \min.$$

Equation (38) can be solved according to the following stages:

1. Cholesky Decomposition of K_1: the kernel function of K_1 is symmetric and positive definite, and can be Cholesky-decomposed:

$$K_1 = VV^T \tag{39}$$

where $V \in R^{N\times N}$ is a low-triangular matrix and the diagonal elements are all positive.

2. Let $Z = YV \in R^{d\times N}$, then $Y = ZV^{-1}$ and

$$\frac{tr\left(YLL^TY^T\right)}{tr\left(YK_1Y^T\right)} = \frac{tr\left(YLL^TY^T\right)}{tr\left(YVV^TY^T\right)} = \frac{tr\left(ZV^{-1}LL^T(V^{-1})^TZ^T\right)}{tr\left(ZZ^T\right)}. \tag{40}$$

Furthermore, let us denote $Z = \begin{bmatrix} Z_{1Row} \\ \vdots \\ Z_{dRow} \end{bmatrix}$, where $Z_{iRow} \in R^{1\times N}$ represents the row vector of Z, $i = 1,\dots,d$, then

$$\frac{tr\left(ZV^{-1}LL^T(V^{-1})^TZ^T\right)}{tr\left(ZZ^T\right)} = \frac{\sum_{i=1}^d Z_{iRow}V^{-1}LL^T(V^{-1})^TZ_{iRow}^T}{\sum_{i=1}^d Z_{iRow}Z_{iRow}^T}. \tag{41}$$

3. Eigen Decomposition of $V^{-1}LL^T(V^{-1})^T$. If Z_{iRow}^T is an eigenvector of $V^{-1}LL^T(V^{-1})^T$, i.e.,

$$V^{-1}LL^T\left(V^{-1}\right)^TZ_{iRow}^T = \lambda_i Z_{iRow}^T \tag{42}$$

then

$$\lambda_{\min} \le \frac{\sum_{i=1}^d Z_{iRow}V^{-1}LL^T(V^{-1})^TZ_{iRow}^T}{\sum_{i=1}^d Z_{iRow}Z_{iRow}^T} = \frac{\sum_{i=1}^d \lambda_i Z_{iRow}Z_{iRow}^T}{\sum_{i=1}^d Z_{iRow}Z_{iRow}^T} \le \lambda_{\max} \tag{43}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ represent the maximum and minimum eigenvalues of $V^{-1}LL^T(V^{-1})^T$, respectively.

It is clear that the d row vectors of Z should be chosen to be the eigenvectors corresponding to the d minimum eigenvalues of $V^{-1}LL^T(V^{-1})^T$.

4. $Y = ZV^{-1}$.

6 EXPERIMENTS

In this section, some experimental results of LTSA and HSIC-LTSA are presented for comparison.

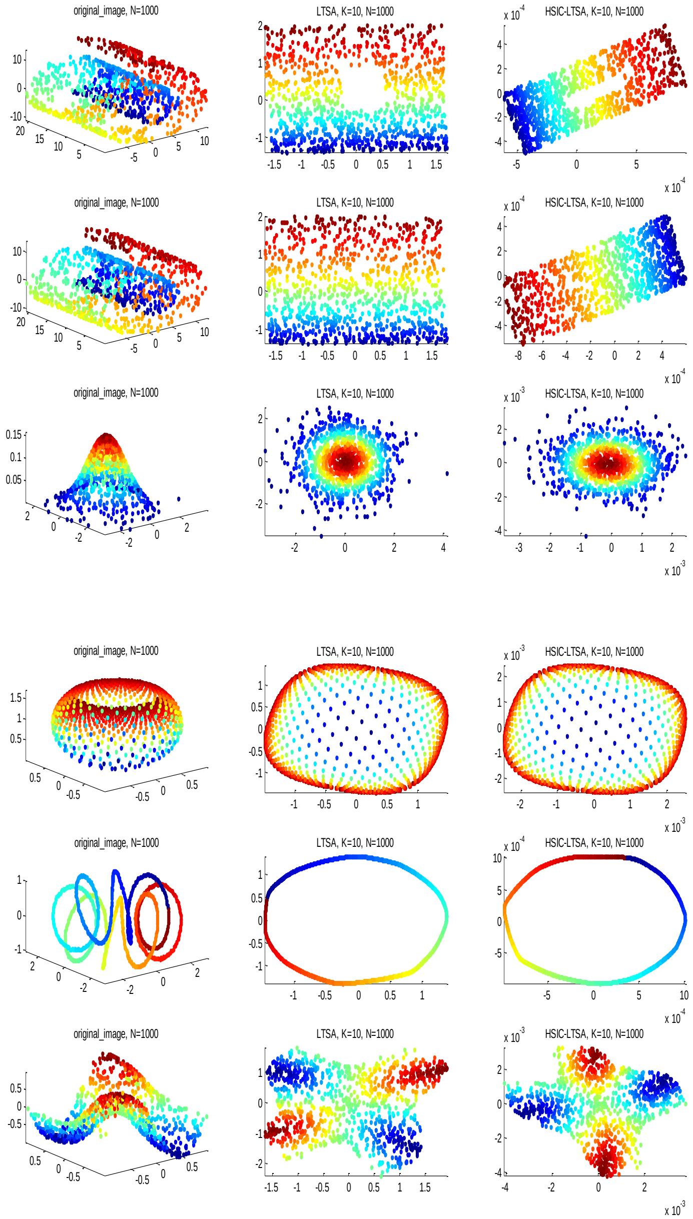

Figure 1. The experimental results of LTSA and HSIC-LTSA on toy data

6.1 Toy Data

Figure 1 shows the experimental results on toy data. The toy data as well as the experimental results of LTSA on the toy data are all produced by using MANI. MANI is a platform commonly used in manifold learning and can be downloaded free from internet. It can be seen from Figure 1 that the experimental results of HSIC-LTSA are reasonable and comparable with those of LTSA. In some toy data, HSIC-LTSA seems even better than LTSA. For example, in Swill Roll with rectangle hole in the middle, HSIC-LTSA reproduces the rectangle more faithfully.

6.2 Face Image Data

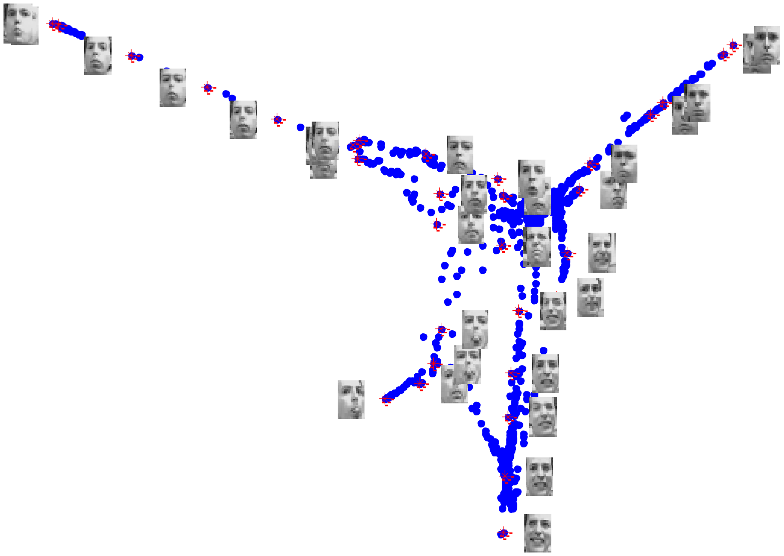

Figure 2. The experimental results of LTSA on face images

Figures 2 and 3 show the experimental results of LTSA and HSIC-LTSA on the dataset of faces. This dataset is often used in many literatures of manifold learning. The face in the dataset only changes in gesture and expression. Therefore, although the faces are represented with high-dimensional vectors, it may be enough to represent these faces with 2-dimensional vectors. In Figures 2 and 3, the faces are dimensionally reduced to 2-dimension plane with LTSA and HSIC-LTSA, respectively. Some face images are also shown at the corresponding positions. It can be seen from Figures 2 and 3 that from up to bottom the face expression

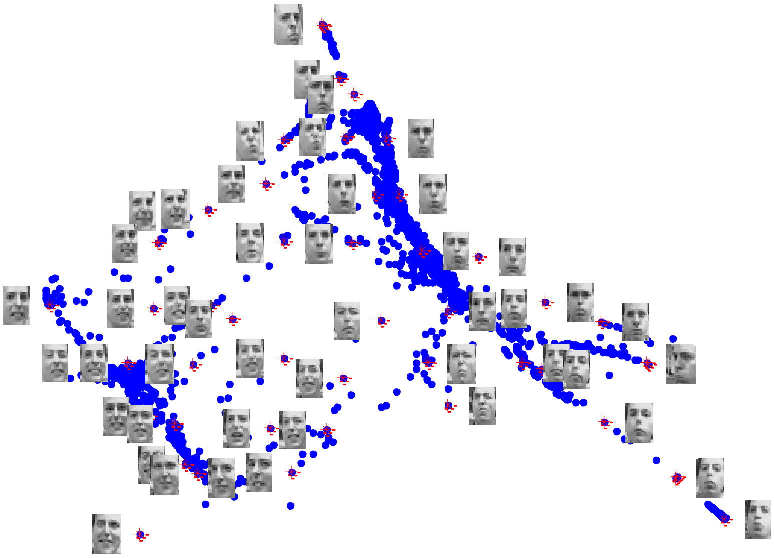

Figure 3. The experimental results of HSIC-LTSA on face images

changes from serious to happy, while from left to right, the face gesture changes from eastward to westward. The impression of HSIC-LTSA seems better than that of LTSA.

6.3 Classification Experiments

The experimental results shown in Figures 1, 2 and 3 are qualitative, not quantitative, and are judged entirely by subjective feelings. In order to compare LTSA and HSIC-LTSA objectively, a number of classification experiments are presented, where data are first dimensionally reduced with LTSA and HSIC-LTSA, respectively, and then classified with K-NN method. The accuracy rates of classification are listed in Table 1.

The datasets used in the classification experiment are MNIST, USPS, YaleB, Binaryalphadigs, AR, UMIST, ORL and Vehicle. All these datasets can be downloaded from Internet and commonly used in many literatures of machine learning. Both MNIST and USPS are the datasets of handwritten digits. Binaryalphadigs is the dataset of handwritten digits and English letters. YaleB, AR, UMIST and ORL are all the datasets of face images. Vehicle is the dataset of vehicle images. The classification method used in the experiments is 3-NN method. The kernel used in HSIC is linear kernel.

In Table 1, the numbers shown in the leftmost column are the reduced dimensions; the numerical values shown next to the names of datasets are the accuracy rates of classification without dimensionality reduction. Since the dimension of feature vectors of vehicle image is only 18, the reduced dimensions are then not larger than 18.

Generally speaking, the performance of HSIC-LTSA is better than LTSA.

RD: the reduced dimension; unit: %

The numbers next to the names of datasets are the accuracy rates of classification without dimensionality reduction

RD	MNIST/88.0		USPS/84.1		YaleB/61.5		AR/32.13	
	LTSA	HSIC-LTSA	LTSA	HSIC-LTSA	LTSA	HSIC-LTSA	LTSA	HSIC-LTSA
10	74.6	86.2	69.4	85.9	7.5	19.5	15.8	17.8
20	80.4	88.6	79.4	87.4	28.0	67.7	20.4	23.0
30	82.4	89.3	80.5	86.4	46.3	78.3	23.4	28.6
40	82.2	88.5	80.6	87.1	61.4	80.5	26.0	31.7
50	85.5	88.7	82.8	86.6	73.6	83.1	28.9	37.3
60	86.0	88.4	82.4	85.4	77.5	83.2	33.7	44.3
80	86.0	88.4	84.7	85.3	82.6	84.7	47.6	51.5
100	87.1	88.1	84.5	84.1	85.3	86.0	58.9	58.3

RD	ORL/82.5		Binaryalphadigs/69.5		RD	Vehicle/63.7	
	LTSA	HSIC-LTSA	LTSA	HSIC-LTSA		LTSA	HSIC-LTSA
10	64.0	77.0	53.1	7.40	2	48.6	48.5
20	75.2	81.8	66.3	31.4	3	48.7	51.3
30	81.9	81.7	65.6	31.4	4	48.0	51.7
40	82.0	77.0	67.4	31.4	5	51.8	50.5
50	81.7	72.5	63.5	43.0	10	66.5	62.3
60	77.6	65.2	59.0	40.0	15	74.0	66.7
80	70.6	53.9	52.4	27.6	16	74.7	70.4
100	66.1	47.4	41.1	18.9	17	75.1	66.9

Remark: The datasets as well as the source codes will be available on request.

Table 1. The accuracy rates of classification

7 CONCLUSIONS

The theory of HSIC sounds a little complicated and seems too difficult to understand for AI engineers. In this paper, a brief and self-sufficient introduction to HSIC is presented for better understanding of HSIC. Since it was first proposed around 2005, HSIC has found many applications in machine learning and some of them are similar to dimensionality reduction [24, 25, 26]. However, HSIC has never been applied to machine learning in regularization form so far. The proposed HSIC-LTSA may be the first try of HSIC regularization.

The so-called regularization means to add regularization terms behind objective functions of other algorithms. The proposed HSIC-LTSA adds HSIC regularization to LTSA, we can also add HSIC regularization to Laplacian Eigenmap algorithm [1] to form HSIC-LE algorithm, to Local Linear Embedded algorithm [2] to form HSIC-LLE algorithm, and so on. HSIC regularization would likely greatly expand the application scope of HSIC, just like what manifold regularization [3] has done. Manifold regularization makes the application scope of manifold learning expand from dimensionality reduction initially to various aspects of machine learning now.

Acknowledgments

This research has been supported by the National Natural Science Foundation of China (NSFC) under Grants 61773022.

REFERENCES

[1] Belkin, M.—Niyogi, P.: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. Proceedings of the 14th International Conference on Neural Information Processing Systems (NIPS 2001). Advances in Neural Information Processing Systems, Vol. 14, 2001, pp. 585–591.

[2] Roweis, S. T.—Saul, L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, Vol. 290, 2000, No. 5500, pp. 2323–2326, doi: 10.1126/science.290.5500.2323.

[3] Belkin, M.—Niyogi, P.—Sindhwani, V.: Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. Journal of Machine Learning Research, Vol. 7, 2006, No. 11, pp. 2399–2434.

[4] Weinberger, K. Q.—Sha, F.—Saul, L. K.: Learning a Kernel Matrix for Nonlinear Dimensionality Reduction. Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004), ACM, International Conference Proceeding Series, Vol. 69, 2004, doi: 10.1145/1015330.1015345.

[5] Lafon, S.—Lee, A. B.: Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, 2006, No. 9, pp. 1393–1403, doi: 10.1109/tpami.2006.184.

[6] Zhang, Z.—Zha, H.: Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. SIAM Journal on Scientific Computing, Vol. 26, 2004, No. 1, pp. 313–338, doi: 10.1137/s1064827502419154.

[7] He, X.—Niyogi, P.: Locality Preserving Projections. Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS 2003). Advances in Neural Information Processing Systems, Vol. 16, 2003, pp. 186–197.

[8] Chen, J.—Ma, Z.—Liu, Z.: Local Coordinates Alignment with Global Preservation for Dimensionality Reduction. IEEE Transactions on Neural Networks and Learning Systems, Vol. 24, 2013, No. 1, pp. 106–117, doi: 10.1109/tnnls.2012.2225844.

[9] Liu, X.—Wang, L.—Zhang, J.—Yin, J.—Liu, H.: Global and Local Structure Preservation for Feature Selection. IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, 2014, No. 6, pp. 1083–1095, doi: 10.1109/tnnls.2013.2287275.

[10] Gretton, A.—Bousquet, O.—Smola, A.—Schölkopf, B.: Measuring Statistical Dependence with Hilbert-Schmidt Norms. In: Jain, S., Simon, H. U., Tomita, E. (Eds.): Algorithmic Learning Theory (ALT 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3734, 2005, pp. 63–77, doi: 10.1007/11564089_7.

[11] Yan, K.—Kou, L.—Zhang, D.: Learning Domain-Invariant Subspace Using Domain Features and Independence Maximization. IEEE Transactions on Cybernetics, Vol. 48, 2018, No. 1, pp. 288–299, doi: 10.1109/tcyb.2016.2633306.

[12] Damodaran, B. B.—Courty, N.—Lefèvre, S.: Sparse Hilbert Schmidt Independence Criterion and Surrogate-Kernel-Based Feature Selection for Hyperspectral Image Classification. IEEE Transactions on Geoscience and Remote Sensing, Vol. 55, 2017, No. 4, pp. 2385–2398, doi: 10.1109/tgrs.2016.2642479.

[13] Gangeh, M. J.—Zarkoob, H.—Ghodsi, A.: Fast and Scalable Feature Selection for Gene Expression Data Using Hilbert-Schmidt Independence Criterion. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), Vol. 14, 2017, No. 1, pp. 167–181, doi: 10.1109/tcbb.2016.2631164.

[14] Xiao, M.—Guo, Y.: Feature Space Independent Semi-Supervised Domain Adaptation via Kernel Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, 2015, No. 1, pp. 54–66, doi: 10.1109/tpami.2014.2343216.

[15] Zhong, W.—Pan, W.—Kwok, J. T.—Tsang, I. W.: Incorporating the Loss Function into Discriminative Clustering of Structured Outputs. IEEE Transactions on Neural Networks, Vol. 21, 2010, No. 10, pp. 1564–1575, doi: 10.1109/tnn.2010.2064177.

[16] Jost, J.: Riemannian Geometry and Geometric Analysis. Springer Science and Business Media, 2008.

[17] Spivak, M.: A Comprehensive Introduction to Differential Geometry, Vol. 4. 3rd edition. Publish or Perish Press, 1999.

[18] Kreyszig, E.: Introductory Functional Analysis with Applications. 1st edition. Wiley, New York, 1989, pp. 547–553.

[19] Mika, S.—Ratsch, G.—Weston, J.—Scholkopf, B.—Mullers, K. R.: Fisher Discriminant Analysis with Kernels. Neural Networks for Signal Processing IX. Proceedings of the 1999 IEEE Signal Processing Society Workshop, 1999, doi: 10.1109/nnsp.1999.788121.

[20] Cortes, C.—Vapnik, V.: Support-Vector Networks. Machine Learning, Vol. 20, 1995, No. 3, pp. 273–297, doi: 10.1007/bf00994018.

[21] Shawe-Taylor, J.—Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, 2004, doi: 10.1017/cbo9780511809682.

[22] Gohberg, I.—Goldberg, S.—Kaashoek, M. A.: Hilbert-Schmidt Operators. Classes of Linear Operators, Vol. I. Birkhäuser, Basel, Operator Theory: Advances and Applications, Vol. 49, 1990, pp. 138–147, doi: 10.1007/978-3-0348-7509-7_9.

[23] Xiang, S.—Nie, F.—Pan, C.—Zhang, C.: Regression Reformulations of LLE and LTSA with Locally Linear Transformation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 41, 2011, No. 5, pp. 1250–1262, doi: 10.1109/tsmcb.2011.2123886.

[24] Gangeh, M. J.—Ghodsi, A.—Kamel, M. S.: Kernelized Supervised Dictionary Learning. IEEE Transactions on Signal Processing, Vol. 61, 2013, No. 19, pp. 4753–4767, doi: 10.1109/tsp.2013.2274276.

[25] Gangeh, M. J.—Fewzee, P.—Ghodsi, A.—Kamel, M. S.—Karray, F.: Multiview Supervised Dictionary Learning in Speech Emotion Recognition. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), Vol. 22, 2014, No. 6, pp. 1056–1068, doi: 10.1109/taslp.2014.2319157.

[26] Barshan, E.—Ghodsi, A.—Azimifar, Z.—Jahromi, M. Z.: Supervised Principal Component Analysis: Visualization, Classification and Regression on Subspaces and Submanifolds. Pattern Recognition, Vol. 44, 2011, No. 7, pp. 1357–1371, doi: 10.1016/j.patcog.2010.12.015.

Xinghua Zheng received his B.Sc., M.Sc. and Ph.D. degrees from the Sun Yat-sen University, Guangzhou, China, in 2006, 2011 and 2018, respectively. His research interests include machine learning.

Zhengming Ma received his B.Sc. and M.Sc. degrees from the South China University of Technology, Guangzhou, China, in 1982 and 1985, respectively, and his Ph.D. degree in pattern recognition and intelligent control from Tsinghua University, Beijing, China, in 1989. He is currently Professor with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. His current research interest is machine learning.

Hangjian Che received his B.Sc. degree in electronic information science and technology from Sun Yat-sen University, Guangzhou, China, in 2017. He is currently pursuing the master's degree with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou. His current research interests include machine learning.

Lei Li received his Ph.D. degree in computer science from Claude Bernard Lyon 1 University, France, in 1988. He is currently Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interest is machine learning.