Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 6791683, 9 pages
https://doi.org/10.1155/2018/6791683

Research Article

Kernel Entropy Component Analysis with Nongreedy L1-Norm Maximization

Haijin Ji[1,2] **and Song Huang**[1]

[1]*Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China*
[2]*School of Computer Science and Technology, Huaiyin Normal University, Huaian 223300, China*

Correspondence should be addressed to Song Huang; hs0317@163.com

Received 13 April 2018; Revised 26 August 2018; Accepted 9 September 2018; Published 14 October 2018

Academic Editor: Pedro Antonio Gutierrez

Kernel entropy component analysis (KECA) is a newly proposed dimensionality reduction (DR) method, which has showed superiority in many pattern analysis issues previously solved by principal component analysis (PCA). The optimized KECA (OKECA) is a state-of-the-art variant of KECA and can return projections retaining more expressive power than KECA. However, OKECA is sensitive to outliers and accused of its high computational complexities due to its inherent properties of L2-norm. To handle these two problems, we develop a new extension to KECA, namely, KECA-L1, for DR or feature extraction. KECA-L1 aims to find a more robust kernel decomposition matrix such that the extracted features retain information potential as much as possible, which is measured by L1-norm. Accordingly, we design a nongreedy iterative algorithm which has much faster convergence than OKECA's. Moreover, a general semisupervised classifier is developed for KECA-based methods and employed into the data classification. Extensive experiments on data classification and software defect prediction demonstrate that our new method is superior to most existing KECA- and PCA-based approaches. Code has been also made publicly available.

1. Introduction

Curse of dimensionality is one of the major issues in machine learning and pattern recognition [1]. It has motivated many scholars from different areas to properly implement dimensionality reduction (DR) to simplify the input space without degrading performances of learning algorithms. Various efficient methods associated with DR have been developed, such as independent component analysis (ICA) [2], linear discriminant analysis [3], principal component analysis (PCA) [4], projection pursuit [5], to name a few. Among these robust algorithms, PCA has been one of the most used techniques to perform feature extraction (or DR). PCA implements linear data transformation according to the projection matrix, which aims to maximize the second-order statistics of input datasets [6]. To extend PCA to nonlinear space, Schölkopf et al. [7] proposed the kernel PCA, the so-called KPCA method. The key of KPCA is to find the nonlinear relation between the input data and the kernel feature space (KFS) using the kernel matrix, which is derived from a positive semidefinite kernel function of computing inner products. Both PCA and KPCA perform data transformation by selecting the eigenvectors corresponding to the top eigenvalues of the projection matrix and the kernel matrix, respectively. All of them (including their variants) have experienced great success in different areas [8–12], such as image reconstruction [13], face recognition [14–17], image processing [18, 19], to name a few. However, as suggested by Zhang and Hancock [20], the DR should be performed according to the perspective of information theory for obtaining more acceptable results.

To improve performances of the aforementioned approaches to DR, Jessen [6] developed a new and completely different data transformation algorithm, namely, kernel entropy component analysis (KECA). The main difference between KECA and PCA or KPCA is that the optimal eigenvectors (or called entropic components) derived from KECA can compress the most Renyi entropy of the input

data instead of being associated with top eigenvalues. The procedure of selecting the eigenvectors related to the Renyi entropy of the input space is started with a Parzen window kernel-based estimator [21]. Then, only the eigenvectors corresponding to the most entropy of the input datasets are selected to perform DR. This distinguished characteristic helps KECA achieve better performances than the classical PCA and KPCA in face recognition and clustering [6]. In recent years, Izquierdo-Verdiguier et al. [21] employed the rotation matrix from ICA [2] to optimize KECA and proposed the optimized KECA (OKECA). OKECA not only shows superiority in classification of both synthetic and real datasets but can obtain acceptable kernel density estimation (KDE) just using very fewer entropic components (just one or two) compared with KECA [21]. However, OKECA is sensitive to outliers for its inherent properties of L2-norm. In other words, if the input space follows normal distribution and is contaminated by nonnormal distributed outliers, this may lead to the downgrade of its performance on DR in terms of OKECA. Additionally, OKECA is very time-consuming when handling large-scale input datasets (Section 4).

Therefore, the main purpose of this paper is to propose a new variant of KECA and improve the proneness to outliers and efficiency of OKECA. L1-norm is well known for its robustness to outliers [22]. Additionally, Nie et al. [23] established a fast iteration process to handle the general L1-norm maximization issue with nongreedy algorithm. Hence, we take advantages of OKECA and propose a new L1-norm version of KECA (denoted as KECA-L1). KECA-L1 uses an efficient convergence procedure, motivated by Nie et al.'s method [23], to search for the entropic components contributing to the most Renyi entropy of input data. To evaluate the efficiency and effectiveness of KECA-L1, we design and conduct a series of experiments, in which the data vary from single class to multiattribute and from small to large size. The classical KECA and OKECA are also included for comparison.

The remainder of this paper is organized as follows: Section 2 reviews the general L1-norm maximization issue, KECA, and OKECA. Section 3 presents KECA with nongreedy L1-norm maximization and semisupervised-learning-based classifier. Section 4 validates the performance of the new method on different data sets. Section 5 ends this paper with some conclusions.

2. Preliminaries

2.1. An Efficient Algorithm to Solving the General L1-Norm Maximization Issue. The general L1-norm maximization problem is first raised by Nie et al. [23]. This issue, based on a hypothesis that there exists an upper bound for the objective function, can be generally formulated as [23]

$$\max_{\nu\in\mathscr{C}} f(\nu) + \sum_i \left|g_i(\nu)\right|, \tag{1}$$

where both $f(\nu)$ and $g_i(\nu)$ for each i denote arbitrary functions, and $\nu \in \mathscr{C}$ represents an arbitrary constraint.

Then a sign function $\mathrm{sign}(\cdot)$ is defined as

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0, \end{cases} \tag{2}$$

and employed to transform the maximization problem (1) as follows:

$$\max_{\nu\in\mathscr{C}} f(\nu) + \sum_i \alpha_i g_i(\nu), \tag{3}$$

where $\alpha_i = \mathrm{sign}(g_i(\nu))$. Nie et al. [23] proposed a fast iteration process to solve problem (3), which is shown in Algorithm 1. It can be seen from Algorithm 1 that α_i is determined by current solution ν^t, and the next solution ν^{t+1} is updated according to the current α_i. The iterative process is repeated until the procedure converges [23, 24]. The convergence of the Algorithm 1 has been demonstrated, and the associated details can also be read in [23].

2.2. Kernel Entropy Component Analysis. KECA is characterized by its entropic components instead of the principal or variance-based components in PCA or KPCA, respectively. Hence, we firstly describe the concept of the Renyi quadratic entropy. Given the input dataset $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ ($\mathbf{x}_i \in \mathbb{R}^D$), the Renyi entropy of $\mathbf{X}$ is defined as [6]

$$H(p) = -\log \int p^2(\mathbf{x}) d\mathbf{x}, \quad \mathbf{x} \in \mathbf{X}, \tag{4}$$

where $p(\mathbf{x})$ is a probability density function. Based on the monotonic property of logarithmic function, Equation (4) can be rewritten as

$$V(p) = \int p^2(\mathbf{x}) d\mathbf{x}. \tag{5}$$

We can estimate Equation (5) using the kernel $k_\sigma(\mathbf{x}, \mathbf{x}_t)$ of Parzen window density estimator determined by the bandwidth coefficient σ [6] such that

$$\begin{aligned} V(p) &\approx \widehat{V}(p) \\ &= \frac{1}{N} \sum_{\mathbf{x}\in\mathbf{X}} p(\mathbf{x}) \\ &= \frac{1}{N} \sum_{\mathbf{x}_i\in\mathbf{X}} \frac{1}{N} \sum_{\mathbf{x}_j\in\mathbf{X}} k_\sigma\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &= \frac{1}{N^2} \mathbf{1}^T \mathbf{K} 1, \end{aligned} \tag{6}$$

where $\mathbf{K}_{ij} = k_\sigma(\mathbf{x}_i, \mathbf{x}_j)$ constitutes the kernel matrix $\mathbf{K}$ and $\mathbf{1}$ represents an N-dimensional vector containing all ones. With the help of the kernel decomposition [6],

$$\mathbf{K} = \mathbf{A}\mathbf{A}^T = \left(\mathbf{E}\mathbf{D}^{1/2}\right)\left(\mathbf{D}^{1/2}\mathbf{E}^T\right). \tag{7}$$

Equation (6) is transformed as follows:

$$\widehat{V}(p) = \frac{1}{N^2} \sum_{i=1}^{N} \left(\sqrt{\lambda_i}\, \mathbf{1}^T \mathbf{e}_i\right)^2, \tag{8}$$

where the diagonal matrix $\mathbf{D}$ and the matrix $\mathbf{E}$ consist of eigenvalues $\lambda_1, \ldots, \lambda_N$ and the corresponding eigenvectors

Initialize $v^1 \in \mathscr{C}$, $t = 1$;
While not converge
  For each i, compute $\alpha_i^t = \text{sign}(g_i(v^t))$;
  $v^{t+1} = \arg\max_{v \in \mathscr{C}} f(v^t) + \sum_i \alpha_i g_i(v^t)$;
  $t = t + 1$;
end
Output: v^{t+1}

Algorithm 1: Fast iteration approach to solving the general L1-Norm maximization problem (3).

$\mathbf{e}_1, \ldots, \mathbf{e}_N$, respectively. It can be observed from Equation (7) that the entropy estimator $\widehat{V}(p)$ consists of projections onto all the KFS axes because

$$\mathbf{K}_{ij} = k_\sigma\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi\left(\mathbf{x}_i\right)^T \phi\left(\mathbf{x}_j\right), \tag{9}$$

where the function of $\phi(\cdot)$ is to map the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ into the KFS. Additionally, only an entropic component $\mathbf{e}_i$ meeting the criteria of $\lambda_i \neq 0$ and $\mathbf{1}^T\mathbf{e}_i \neq 0$ can contribute to the entropy estimate [21]. In a word, KECA implements DR by projecting $\phi(\mathbf{X})$ into a subspace $\mathbf{E}_l$ spanned not by the eigenvectors associated with the top eigenvalues but by entropic components contributing most to the Renyi entropy estimator $\widehat{V}(p)$ [25].

2.3. Optimized Kernel Entropy Component Analysis. Due to the fact that KECA is sensitive to different bandwidth coefficients σ [21], OKECA is proposed to fill this gap and improve performances of KECA on DR. Motivated by the fast ICA method [2], an extra rotation matrix (applying $\mathbf{W}$) is employed to the kernel decomposition (Equation (7)) in KECA for maximizing the information potential (the entropy values in Equation (8)) [21]:

$$\begin{cases} \max\limits_{\mathbf{w}_k \in \mathbf{W}} & J(\mathbf{w}) = \left(\mathbf{1}_N^T \mathbf{E}\mathbf{D}^{1/2}\mathbf{w}\right)^2, \\ \text{s.t.} & \mathbf{W}\mathbf{W}^T = \mathbf{I}, \ \|\mathbf{w}\|_2 = 1, \end{cases} \tag{10}$$

where $\|\cdot\|_2$ is the L2-norm and $\mathbf{w}$ denotes a column vector $(N \times 1)$ in $\mathbf{W}$. Izquierdo-Verdiguier et al. [21] utilized a gradient-ascent approach to handle the maximization problem (10):

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \tau \frac{\partial J}{\partial \mathbf{w}(t)}, \tag{11}$$

where τ is the step size. $\partial J/\partial \mathbf{w}(t)$ can be obtained by Lagrangian multiplier:

$$\frac{\partial J}{\partial \mathbf{w}(t)} = \frac{\partial \mathscr{L}(\mathbf{w})}{\partial \mathbf{w}} = 2\left(\mathbf{1}_N^T \mathbf{E}\mathbf{D}^{1/2}\mathbf{w}\right)\left(\mathbf{1}_N^T \mathbf{E}\mathbf{D}^{1/2}\right)^T. \tag{12}$$

The entropic components multiplied by the rotation matrix can obtain more (or equal) information potential than that of the KECA even using fewer components [21]. Moreover, OKECA shows the capability of being robust to the bandwidth coefficient. However, there exist two main limitations for OKECA. First, the new entropic components derived from OKECA are sensible to outliers since its inherent properties of L2-norm (Equation (10)). Second, although a very simple stopping criterion is designed to avoid additional iterations, OKECA is still of high computational complexities for its computational cost is $O(N^3 + 4tN^2)$ [21], where t is the number of iterations for finding the optimal rotation matrix, compared with that the one of KECA is $O(N^3)$ [21].

3. KECA with Nongreedy L1-Norm Maximization

3.1. Algorithm. In order to alleviate the problems existing in OKECA, this section presents how to extend KECA to its nongreedy L1-norm version. For readers' easy understanding, the definition of L1-norm is firstly introduced as follows:

Definition 1. Given an arbitrary vector $\mathbf{x} \in \mathbb{R}^{N\times 1}$, the L1-norm of the vector $\mathbf{x}$ is

$$\|\mathbf{x}\|_1 = \sum_{j=1}^{N} \left|x_j\right|, \tag{13}$$

where $\|\cdot\|_1$ is the L1-norm and x_j denotes the jth element of $\mathbf{x}$.

Then, motivated by OKECA, we attempt to develop a new objective function to maximize the information potential (Equations (8) and (10)) based on the L1-norm:

$$\begin{cases} \max & J(\mathbf{W}) = \left\|\mathbf{W}^T\mathbf{E}\mathbf{D}^{1/2}\right\|_1 = \sum\limits_{j=1}^{N} \text{sign}\left(\mathbf{a}_j^T\mathbf{W}\right)\mathbf{W}^T\mathbf{a}_j, \\ \text{s.t.} & \mathbf{W}^T\mathbf{W} = \mathbf{I}, \end{cases} \tag{14}$$

where $(\mathbf{a}_1, \ldots, \mathbf{a}_N) = \mathbf{A} = \mathbf{E}\mathbf{D}^{1/2}$, N is the size of samples. The rotation matrix is denoted as $\mathbf{W} \in \mathbb{R}^{\text{DIM}\times m}$, where DIM and m are the dimension of input data and dimension of the selected entropic components (or number of projection), respectively. It is difficult to directly solve problem (14), but we may regard it as a special case of problem (1) when $f(v) \equiv 0$. Therefore, the Algorithm 1 can be employed to solve (14). Next, we show the details about how to find the optimal solution of problem (14) based on the proposal from References [23, 24]. Let

$$\mathbf{M} = \sum_{j=1}^{N} \mathbf{a}_j \text{sign}\left(\mathbf{a}_j^T\mathbf{W}\right). \tag{15}$$

Thus, problem (14) can be simplified as

$$\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \quad J(\mathbf{w}) = \text{Tr}\left(\mathbf{W}^T\mathbf{M}\right). \tag{16}$$

By singular value decomposition (SVD), then

$$\mathbf{M} = \mathbf{U}\Lambda\mathbf{V}^T, \tag{17}$$

where $\mathbf{U} \in \mathbb{R}^{\text{DIM}\times\text{DIM}}$, $\Lambda \in \mathbb{R}^{\text{DIM}\times m}$, and $\mathbf{V} \in \mathbb{R}^{m\times m}$. Then we obtain

Input $\mathbf{K}$ and m
Initialize $\mathbf{W}^1 \in \mathbb{R}^{\mathrm{DIM}\times m}$ such that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, $t = 1$;
/----------Phase 1----------/
(1) Eigen decomposition. $[\mathbf{E}, \mathbf{D}] \longleftarrow \mathrm{eig}(\mathbf{K})$. $\mathbf{D} \longleftarrow \mathrm{sort}(\mathbf{D})$, $\mathbf{E} \longleftarrow \mathrm{sort}(\mathbf{E})$, $\mathbf{A} = \mathbf{E}\mathbf{D}^{1/2}$;
/----------Phase 2----------/
While not converge
(2) $\mathbf{M} = \sum_{j=1}^{N} \mathbf{a}_j \mathrm{sign}(\mathbf{a}_j^T\mathbf{W})$;
(3) Compute the SVD of $\mathbf{M}$ as $\mathbf{M} = \mathbf{U}\Lambda\mathbf{V}^T$. Let $\mathbf{W}^{t+1} = \mathbf{U}[\mathbf{I}_m; \mathbf{0}_{(\mathrm{DIM}-m)\times m}]\mathbf{V}^T$;
(4) $t = t + 1$;
end
Output: $\mathbf{W}^{t+1} \in \mathbb{R}^{\mathrm{DIM}\times m}$

Algorithm 2: KECA-L1.

$$\begin{aligned}\mathrm{Tr}\left(\mathbf{W}^T\mathbf{M}\right) &= \mathrm{Tr}\left(\mathbf{W}^T\mathbf{U}\Lambda\mathbf{V}^T\right) = \mathrm{Tr}\left(\Lambda\mathbf{V}^T\mathbf{W}^T\mathbf{U}\right) \\ &= \mathrm{Tr}(\Lambda\mathbf{Z}) = \sum_i \lambda_{ii} z_{ii},\end{aligned} \tag{18}$$

where $\mathbf{Z} \in \mathbb{R}^{m\times m}$, λ_{ii} and z_{ii} denote the $(i,i)-$th element of matrix Λ and $\mathbf{Z}$, respectively. Due to the property of SVD, we have $\lambda_{ii} \geq 0$. Additionally, $\mathbf{Z}$ is an orthonormal matrix [23] such that $z_{ii} \leq 1$. Therefore, $\mathrm{Tr}(\mathbf{W}^T\mathbf{M})$ can reach the maximum only if $\mathbf{Z} = [\mathbf{I}_m, \mathbf{0}_{m\times(\mathrm{DIM}-m)}]$, where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix, and $\mathbf{0}_{m\times(\mathrm{DIM}-m)}$ is a $m \times (\mathrm{DIM} - m)$ matrix of zeros. Considering that $\mathbf{Z} = \mathbf{V}^T\mathbf{W}^T\mathbf{U}$, thus the solution to problem (16) is

$$\mathbf{W} = \mathbf{U}\left[\mathbf{I}_m; \mathbf{0}_{(\mathrm{DIM}-m)\times m}\right]\mathbf{V}^T. \tag{19}$$

Algorithm 2 (A MATLAB implementation of the algorithm is available at the Supporting Document for the interested readers) shows how to utilize the nongreedy L1-norm maximization described in Algorithm 1 to compute Equation (19). Since problem (16) is a special case of problem (1), we can obviously obtain that the optimal solution $\mathbf{W}^*$ to Equation (19) is a local maximum point for $\|\mathbf{W}^T\mathbf{E}\mathbf{D}^{1/2}\|_1$ based on Theorem 2 in Reference [23]. Moreover, the Phase 1 of the Algorithm 2 spends $O(N^3)$ on the eigen decomposition. Thus, the total of computational cost of KECA-L1 is $O(N^3 + Nt)$, where t is the number of iterations for convergence. Considering that the computational complexity of OKECA is $O(N^3 + 4tN^2)$, we can safely conclude that KECA-L1 has much faster convergence than OKECA's.

3.2. The Convergence Analysis. This subsection attempts to demonstrate the convergence of the Algorithm 2 in the following: theorem:

Theorem 1. *The above KECA-L1 procedure can converge.*

Proof. Motivated by References [23, 24], first we show the objective function (9) of KECA-L1 will monotonically increase in each iteration t. Let $g_i(u^t) = \mathbf{W}^T\mathbf{a}_j$ and $\alpha_i^t = \mathrm{sign}(\mathbf{a}_j^T\mathbf{W})$, then (9) can be simplified to

Table 1: UCI datasets description.

Database	N	DIM	N_c	N_{train}	N_{test}
Ionosphere	351	33	2	30×2	175
Letter	20000	16	26	35×26	3870
Pendigits	10992	16	9	60×9	3500
Pima-Indians	768	8	2	100×2	325
WDBC	569	30	2	35×2	345
Wine	178	12	3	30×3	80

N: number of samples, DIM: number of dimensions, N_c: number of classes, N_{train}: number of training data, and N_{test}: number of testing data.

$$\begin{cases} \max & J(\mathbf{W}) = \sum_{j=1}^{N} \mathrm{sign}\left(\mathbf{a}_j^T\mathbf{W}\right)\mathbf{W}^T\mathbf{a}_j = \sum_{j=1}^{N} \alpha_i^t g_i\left(u^t\right), \\ \text{s.t.} & \mathbf{W}^T\mathbf{W} = \mathbf{I}. \end{cases} \tag{20}$$

Obviously, α_i^{t+1} is parallel to $g_i(u^{t+1})$, but neither is α_i^t. Therefore,

$$\begin{aligned}\left|g_i\left(u^{t+1}\right)\right| = \alpha_i^{t+1} g_i\left(u^{t+1}\right) &\geq \alpha_i^t g_i\left(u^{t+1}\right) \\ \Rightarrow \left|g_i\left(u^{t+1}\right)\right| - \alpha_i^t g_i\left(u^{t+1}\right) &\geq 0.\end{aligned} \tag{21}$$

Considering that $|g_i(u^t)| = \alpha_i^t g_i(u^t)$, thus

$$\left|g_i\left(u^t\right)\right| - \alpha_i^t g_i\left(u^t\right) = 0. \tag{22}$$

Substituting (22) in (21), it can be obtained

$$\left|g_i\left(u^{t+1}\right)\right| - \alpha_i^t g_i\left(u^{t+1}\right) \geq \left|g_i\left(u^t\right)\right| - \alpha_i^t g_i\left(u^t\right). \tag{23}$$

According to the Step 3 in Algorithm 2 and the theory of SVD, for each iteration t, we have

$$\sum_{i=1}^{N} \alpha_i^t g_i\left(u^{t+1}\right) \geq \sum_{i=1}^{N} \alpha_i^t g_i\left(u^t\right). \tag{24}$$

Combining (23) and (24) for every i, we have

$$\begin{aligned}\sum_{i=1}^{N}\left(\left|g_i\left(u^{t+1}\right)\right| - \alpha_i^t g_i\left(u^{t+1}\right)\right) &\geq \sum_{i=1}^{N}\left(\left|g_i\left(u^t\right)\right| - \alpha_i^t g_i\left(u^t\right)\right) \\ \Rightarrow \sum_{i=1}^{N}\left|g_i\left(u^{t+1}\right)\right| &\geq \sum_{i=1}^{N}\left|g_i\left(u^t\right)\right|,\end{aligned} \tag{25}$$

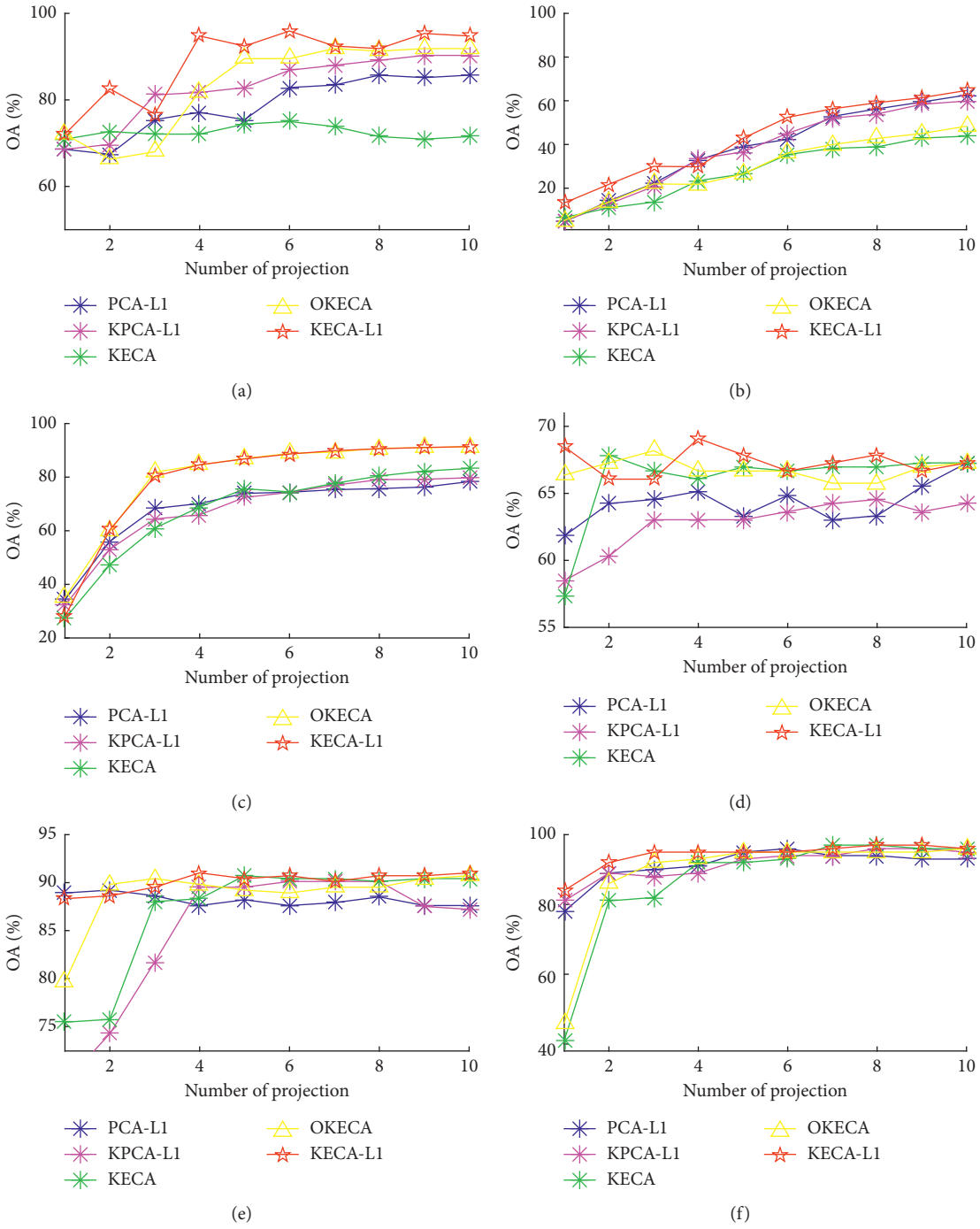

Figure 1: Overall accuracy obtained by the PCA-L1, KPCA-L1, KECA, OKECA, and KECA-L1 using different UCI databases with different numbers of extracted features. (a) Ionosphere, (b) Letter, (c) Pendigits, (d) Pima-Indians, (e) WDBC, and (f) Wine.

which means that Algorithm 2 is monotonically increasing. Additionally, considering that objective function (14) of KECA-L1 has an upper bound within the limited iterations, the KECA-L1 procedure will converge. □

3.3. The Semisupervised Classifier. Jenssen [26] established a semisupervised learning (SSL) algorithm for classification using KECA. This SSL-based classifier was trained by both labeled and unlabeled data to build the kernel matrix such that it can map the data to KFS appropriately [26]. Additionally, it is based on a general modelling scheme and applicable for other variants of KECA, such as OKECA and KECA-L1.

More specifically, we are given N pairs of training data $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ with samples $\mathbf{x}_i \in \mathbb{R}^D$ and the associated labels y_i. In addition, there are M unlabeled data points for testing. Let $\mathbf{X}_u = [\mathbf{x}_u^1, \ldots, \mathbf{x}_u^M]$ and $\mathbf{X}_l = [\mathbf{x}_l^1, \ldots, \mathbf{x}_l^N]$ denote the testing data and training data without labels, respectively;

thus, we can obtain an overall matrix $\mathbf{X} = [\mathbf{X}_u\ \mathbf{X}_l]$. Then we construct the kernel matrix $\mathbf{K}$ derived from $\mathbf{X}$ using (6), $\mathbf{K} \in \mathbb{R}^{(N+M)\times(N+M)}$, which plays as the input of Algorithm 2. After the iteration procedure of nongreedy L1-norm maximization, we obtain a projection of $\mathbf{X} \longrightarrow \widetilde{E} = [\widetilde{E}_u \widetilde{E}_l]_{m\times(M+N)}$ onto m orthogonal axes, where $\widetilde{E}_u = [\widetilde{e}_1^u, \ldots, \widetilde{e}_M^u]$ and $\widetilde{E}_l = [\widetilde{e}_1^l, \ldots, \widetilde{e}_N^l]$. In other words, $\widetilde{e}_i^u$ and $\widetilde{e}_j^l$ are the low-dimensional representations of each testing data point $\mathbf{x}_u^i$ and the training one $\mathbf{x}_l^j$, respectively. Assume that $\mathbf{x}_u^*$ is an arbitrary data point to be tested. If it satisfies

$$\left\|\widetilde{e}_u^* - \widetilde{e}_l^j\right\|_2 = \min_{h=1,\ldots,N} \left\|\widetilde{e}_u^* - \widetilde{e}_l^h\right\|_2, \tag{26}$$

then $\mathbf{x}_u^*$ is assigned to the same class with the jth data point of $\mathbf{X}_l$.

4. Experiments

This section shows the performance of the proposed KECA-L1 compared with the classical KECA [6] and OKECA [21] for real-world data classification using the SSL-based classifier illustrated in Section 3.3. Several recent techniques such as PCA-L1 [27] and KPCA-L1 [28] are also included for comparison. The rationale to select these methods is that previous studies related to DR found that they can produce impressive results [27–29]. We implement the experiments on a wide range of real-world datasets: (1) six different datasets from the University California Irvine (UCI) Machine Learning Repository (available at http://archive.ics.uci.edu/ml/datasets.html) and (2) 9 different software projects with 34 releases from the PROMISE data repository (available at http://openscience.us/repo). The MATLAB source code for running KECA and OKECA, uploaded by Izquierdo-Verdiguier et al. [21], is available at http://isp.uv.es/soft_feature.html. The coefficients set for PCA-L1 and KPCA-L1 is the same with [27, 28]. All of the experiments are all performed by MATLAB R2012a on a PC with Inter Core i5 CPU, 4 GB memory, and Windows 7 operating system.

4.1. Experiments on UCI Datasets. The experiments are conducted on six datasets from the UCI: the Inonosphere dataset is a binary classification problem of whether the radar signal can describe the structure of free electrons in the ionosphere or not; the Letter dataset is to assign each black-and-white rectangular pixel display to one of the 26 capital letters in the English alphabet; the Pendigits handles the recognition of pen-based handwritten digits; the Pima-Indians data set constitutes a clinical problem of diabetes diagnosis in patients from clinical variables; the WDBC dataset is another clinical problem for the diagnosis of breast cancer in malignant or benign classes; and the Wine dataset is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. Table 1 shows the details of them. In the subsequent experiments, we just utilized the simplest linear classifier [30]. The theory of maximizing maximum likelihood (ML) [31] is selected as the rule for selecting bandwidth coefficient as suggested in [21].

Table 2: Descriptions of data attributes.

Attribute	Description
WMC	Weighted methods per class
AMC	Average method Complexity
AVG_CC	Mean values of methods in the same class
CA	Afferent couplings
CAM	Cohesion among methods of class
CBM	Coupling between Methods
CBO	Coupling between object classes
CE	Efferent couplings
DAM	Data access Metric
DIT	Depth of inheritance tree
IC	Inheritance Coupling
LCOM	Lack of cohesion in Methods
LCOM3	Normalized version of LCOM
LOC	Lines of code
MAX_CC	Maximum values of methods in the same class
MFA	Measure of function Abstraction
MOA	Measure of Aggregation
NOC	Number of Children
NPM	Number of public Methods
RFC	Response for a class
Bug	Number of bugs detected in the class

Table 3: Descriptions of software data.

Releases	#Classes	#FP	% FP
Ant-1.3	125	20	0.160
Ant-1.4	178	40	0.225
Ant-1.5	293	32	0.109
Ant-1.6	351	92	0.262
Ant-1.7	745	166	0.223
Camel-1.0	339	13	0.038
Camel-1.2	608	216	0.355
Camel-1.4	872	145	0.166
Camel-1.6	965	188	0.195
Ivy-1.1	111	63	0.568
Ivy-1.4	241	16	0.066
Ivy-2.0	352	40	0.114
Jedit-3.2	272	90	0.331
Jedit-4.0	306	75	0.245
Lucene-2.0	195	91	0.467
Lucene-2.2	247	144	0.583
Lucene-2.4	340	203	0.597
Poi-1.5	237	141	0.595
Poi-2.0	314	37	0.118
Poi-2.5	385	248	0.644
Poi-3.0	442	281	0.636
Synapse-1.0	157	16	0.102
Synapse-1.1	222	60	0.270
Synapse-1.2	256	86	0.336
Synapse-1.4	196	147	0.750
Synapse-1.5	214	142	0.664
Synapse-1.6	229	78	0.341
Xalan-2.4	723	110	0.152
Xalan-2.5	803	387	0.482
Xalan-2.6	885	411	0.464
Xerces-init	162	77	0.475
Xerces-1.2	440	71	0.161
Xerces-1.3	453	69	0.152
Xerces-1.4	588	437	0.743

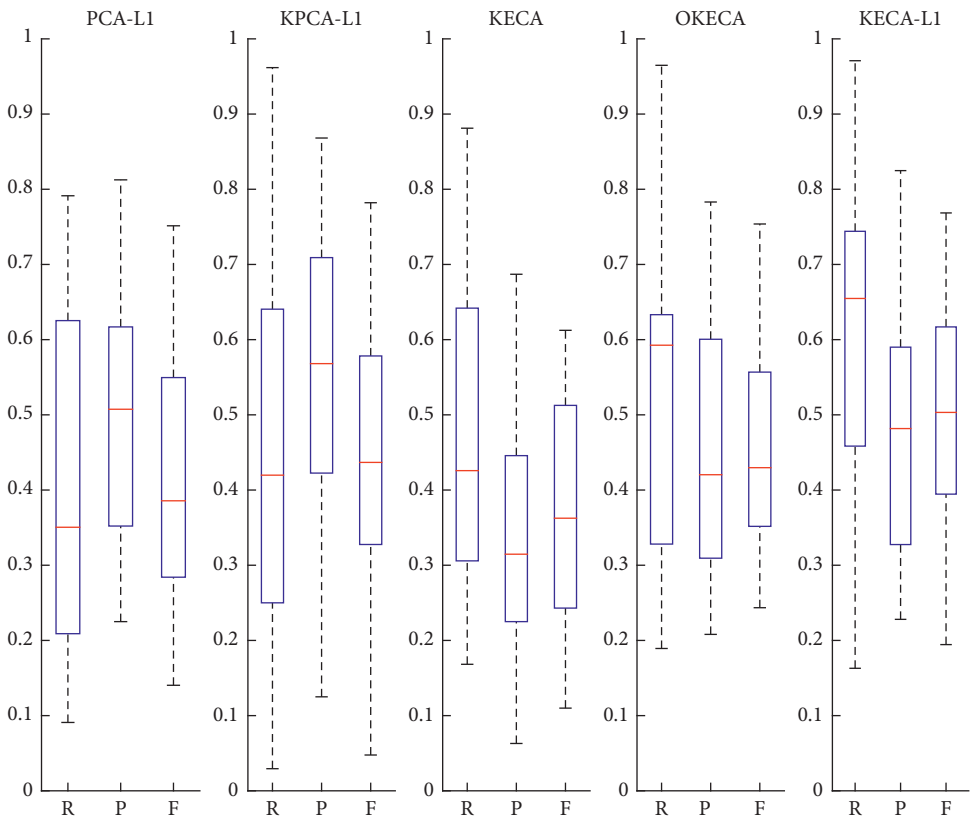

Figure 2: The standardized boxplots of the performance achieved by PCA-L1, KPCA-L1, KECA, OKECA, and KECA-L1, respectively. From the bottom to the top of a standardized box plot: minimum, first quartile, median, third quartile, and maximum.

The implementation of KECA-L1 and other methods is repeated using all the selected datasets with respect to different numbers of components for 10 times. We have utilized the overall classification accuracy (OA) to evaluate the performance of different algorithms on the classification. OA is defined as the total number of samples correctly assigned in percentage terms, which is within $[0, 1]$ and indicates better quality with larger values. Figure 1 presents the average OA curves obtained by the aforementioned algorithms for these six real datasets. It can be observed from Figure 1 that OKECA is superior to KECA, PCA-L1, and KPCA-L1 except for solving Letter issue. This is probably because DR performed by OKECA not only can reveal the structure related to the most Renyi entropy of the original data but also consider the rotational invariance property [21]. In addition, KECA-L1 outperforms the other methods besides of OKECA. This may be attributed to the robustness of L1-norm to outliers compared with that of the L2-norm. In Figure 1(c), OKECA seems to obtain nearly the same results with KECA-L1's. However, the average running time (in hours) of OKECA in the Pendigits is 37.384 times more than that of KECA-L1 1.339.

4.2. Experiments on Software Projects. In software engineering, it is usually difficult to test a software project completely and thoroughly with the limited resources [32]. Software defect prediction (SDP) may provide a relatively acceptable solution to this problem. It can allocate the limited test resources effectively by categorizing the software modules into two classes: nonfault-prone (NFP) or fault-prone (FP) according to 21 software metrics (Table 2).

This section aims to employ KECA-based methods to reduce the selected software data (Table 3) dimensions and then utilize the SSL-based classifier combined with the support vector machine [33] to classify each software module as NFP or FP. The bandwidth coefficient set is still restricted to the rule of ML. PCA-L1 and KPCA-L1 are involved as a benchmarking yardstick. There are 34 groups of tests for each release in Table 3. The most suitable releases [34] from different software projects are selected as training data. We evaluate the performance of different selected methods on SDP in terms of recall (R), precision (P), and F-measure (F) [35, 36]. The F-measure is defined as

$$F = \frac{2 \times \text{precsion} \times \text{recall}}{\text{precsion} + \text{recall}}, \tag{27}$$

where

$$\begin{aligned} \text{Precsion} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \tag{28}$$

In (28), FN (i.e., false negative) means that buggy classes are wrongly classified to be nonfaulty, while FP (i.e., false positive) means nonbuggy classes are wrongly classified to be faulty. TP (i.e., true positive) refer to

correctly classified buggy classes [34]. Values of Recall, Precision, and F-measure range from 0 to 1 and higher values indicate better classification results.

Figure 2 shows the results using box-plot analysis. From Figure 2, considering the minimum, maximum, median, first quartile, and third quartile of the boxes, we find that KECA-L1 performs better than the other methods in general. Specifically, KECA-L1 can obtain acceptable results in experiments for SDP compared with the benchmarks proposed in Reference [34], since the median values of the boxes with respect to R and F are close to 0.7 and more than 0.5, respectively. On the contrary, not only KECA and OKECA but PCA-L1 and KPCA-L1 cannot meet these criteria. Therefore, all of the results validate the robustness of KECA-L1.

5. Conclusions

This paper proposes a new extension to the OKECA approach for dimensional reduction. The new method (i.e., KECA-L1) employs L1-norm and a rotation matrix to maximize information potential of the input data. In order to find the optimal entropic kernel components, motivated by Nie et al.'s algorithm [23], we design a nongreedy iterative process which has much faster convergence than OKECA's. Moreover, a general semisupervised learning algorithm has been established for classification using KECA-L1. Compared with several recently proposed KECA- and PCA-based approaches, this SSL-based classifier can remarkably promote the performance on real-world datasets classification and software defect prediction.

Although KECA-L1 has achieved impressive success on real examples, several problems still should be considered and solved in the future research. The efficiency of KECA-L1 has to be optimized for it is relatively time-consuming compared with most existing PCA-based methods. Additionally, the utilization of KECA-L1 is expected to appear in each pattern analysis algorithm previously based on PCA approaches.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61702544) and Natural Science Foundation of Jiangsu Province of China (Grant no. BK20160769).

Supplementary Materials

The MATLAB toolbox of KECA-L1 is available. (*Supplementary Materials*)

References

[1] Q. Gao, S. Xu, F. Chen, C. Ding, X. Gao, and Y. Li, "R_1-2-DPCA and face recognition," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–12, 2018.

[2] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *European Conference on Computer Vision*, vol. 1, pp. 43–58, 1996.

[4] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[5] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. 23, no. 9, pp. 881–890, 1974.

[6] R. Jenssen, "Kernel entropy component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 847–860, 2010.

[7] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[8] S. Mika, A. Smola, and M. Scholz, "Kernel PCA and denoising in feature spaces," *Conference on Advances in Neural Information Processing Systems II*, vol. 11, pp. 536–542, 1999.

[9] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[10] K. Nishino, S. K. Nayar, and T. Jebara, "Clustered blockwise PCA for representing visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1675–1679, 2005.

[11] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 506–513, Washington, DC, USA, June-July 2004.

[12] A. D'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007.

[13] M. Luo, F. Nie, X. Chang, Y. Yang, A. Hauptmann, and Q. Zheng, "Avoiding optimal mean robust PCA/2DPCA with non-greedy L1-norm maximization," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1802–1808, New York, NY, USA, July 2016.

[14] Q. Yu, R. Wang, X. Yang, B. N. Li, and M. Yao, "Diagonal principal component analysis with non-greedy L1-norm maximization for face recognition," *Neurocomputing*, vol. 171, pp. 57–62, 2016.

[15] B. N. Li, Q. Yu, R. Wang, K. Xiang, M. Wang, and X. Li, "Block principal component analysis with nongreedy L1-norm maximization," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2543–2547, 2016.

[16] F. Nie and H. Huang, "Non-greedy L21-norm maximization for principal component analysis," 2016, http://arxiv.org/abs/1603.08293v1.

[17] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *Proceedings of International*

Conference on Machine Learning, pp. 1062–1070, Beijing, China, June 2014.

[18] R. Wang, F. Nie, R. Hong, X. Chang, X. Yang, and W. Yu, "Fast and orthogonal locality preserving projections for dimensionality reduction," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 5019–5030, 2017.

[19] C. Zhang, F. Nie, and S. Xiang, "A general kernelization framework for learning algorithms based on kernel PCA," *Neurocomputing*, vol. 73, no. 4–6, pp. 959–967, 2010.

[20] Z. Zhang and E. R. Hancock, "Kernel entropy-based unsupervised spectral feature selection," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 5, article 1260002, 2012.

[21] E. Izquierdo-Verdiguier, V. Laparra, R. Jenssen, L. Gomez-Chova, and G. Camps-Valls, "Optimized kernel entropy components," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1466–1472, 2017.

[22] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Transactions on Systems Man and Cybernetics Part B*, vol. 40, no. 4, pp. 1170–1175, 2010.

[23] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, "Robust principal component analysis with non-greedy L1-norm maximization," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1433–1438, Barcelona, Catalonia, Spain, July 2011.

[24] R. Wang, F. Nie, X. Yang, F. Gao, and M. Yao, "Robust 2DPCA with non-greedy L1-norm maximization for image analysis," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1108–1112, 2015.

[25] B. H. Shekar, M. Sharmila Kumari, L. M. Mestetskiy, and N. F. Dyshkant, "Face recognition using kernel entropy component analysis," *Neurocomputing*, vol. 74, no. 6, pp. 1053–1057, 2011.

[26] R. Jenssen, "Kernel entropy component analysis: new theory and semi-supervised learning," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, Beijing, China, September 2011.

[27] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008.

[28] Y. Xiao, H. Wang, W. Xu, and J. Zhou, "L1 norm based KPCA for novelty detection," *Pattern Recognition*, vol. 46, no. 1, pp. 389–396, 2013.

[29] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of gaussian kernel for one-class SVM," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 941–953, 2015.

[30] W. Krzanowski, *Principles of Multivariate Analysis*, Vol. 23, Oxford University Press (OUP), Oxford, UK, 2000.

[31] Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Transactions on Computers*, vol. 25, no. 11, pp. 1175–1179, 1976.

[32] W. Liu, S. Liu, Q. Gu, J. Chen, X. Chen, and D. Chen, "Empirical studies of a two-stage data preprocessing approach for software fault prediction," *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 38–53, 2016.

[33] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification models for software defect prediction: a proposed framework and novel findings," *IEEE Transactions on Software Engineering*, vol. 34, no. 4, pp. 485–496, 2008.

[34] Z. He, F. Shu, Y. Yang, M. Li, and Q. Wang, "An investigation on the feasibility of cross-project defect prediction," *Automated Software Engineering*, vol. 19, no. 2, pp. 167–199, 2012.

[35] P. He, B. Li, X. Liu, J. Chen, and Y. Ma, "An empirical study on software defect prediction with a simplified metric set," *Information and Software Technology*, vol. 59, pp. 170–190, 2015.

[36] Y. Wu, S. Huang, H. Ji, C. Zheng, and C. Bai, "A novel Bayes defect predictor based on information diffusion function," *Knowledge-Based Systems*, vol. 144, pp. 1–8, 2018.