Robust Learning with Kernel Mean p-Power Error Loss

Badong Chen, *Senior Member, IEEE*, Lei Xing, *Student Member, IEEE*, Xin Wang, *Student Member, IEEE*, Jing Qin, *Member, IEEE*, Nanning Zheng, *Fellow, IEEE*

***Abstract*—Correntropy is a second order statistical measure in kernel space, which has been successfully applied in robust learning and signal processing. In this paper, we define a non-second order statistical measure in kernel space, called the *kernel mean-p power error* (KMPE), including the *correntropic loss* (C-Loss) as a special case. Some basic properties of KMPE are presented. In particular, we apply the KMPE to *extreme learning machine* (ELM) and *principal component analysis* (PCA), and develop two robust learning algorithms, namely ELM-KMPE and PCA-KMPE. Experimental results on synthetic and benchmark data show that the developed algorithms can achieve consistently better performance when compared with some existing methods.**

Key Words: Robust learning; kernel mean p-power error; extreme learning machine; principal component analysis.

I. Introduction

THE basic framework in learning theory generally considers learning from examples by optimizing (minimizing or maximizing) a certain loss function such that the learned model can discover the structures (or dependencies) in the data generating system under the uncertainty caused by noise or unknown knowledge about the system [1]. The second order statistical measures such as mean square error (MSE), variance and correlation and have been commonly used as the loss functions in machine learning or adaptive system training due to their simplicity and mathematical tractability. For example, the goal of the *least squares* (LS) regression is to learn an unknown mapping (linear or nonlinear) such that MSE between the model output and desired response is minimized. Also, the orthogonal linear transformation in *principal component analysis* (PCA) is determined such that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components [2]. The *canonical-correlation analysis* (CCA) is another example, where the goal is to find the linear combinations of the components in two random vectors which have maximum correlation with each other [3].

The loss functions based on the second order statistical measures, however, are sensitive to outliers in the data, and are not good solution to learning with non-Gaussian data in general [1]. To handle non-Gaussian data (or noises), various non-second order (or non-quadratic) loss functions are frequently applied to learning systems. Typical examples include Huber's min-max loss [4], [5], Lorentzian error loss [5], risk-sensitive loss [6] and mean p-power error (MPE) loss [7], [8]. The MPE is the p-th absolute moment of the error, which with a proper p value can deal with non-Gaussian data well. In general, MPE is robust to large outliers when $p < 2$ [7]. Information theoretic measures, such as entropy, KL divergence and mutual information can also be used as loss functions in machine learning and non-Gaussian signal processing since they can capture higher order statistics (i.e. moments or correlations beyond second order) of the data [1]. Many numerical examples have shown the superior performance of *information theoretic learning*(ITL) [1], [9]. Particularly in recent years, a novel ITL similarity measure, called correntropy, has been successfully applied to robust learning and signal processing [10]–[18]. Correntropy is a generalized correlation in high dimensional kernel space (usually induced by a Gaussian kernel), which is directly related to the probability of how similar two random variables are in a neighborhood (controlled by the kernel bandwidth) of the joint space [10]. Since correntropy is a local similarity measure, it can increase the robustness with respect to outliers by assigning small weights to data beyond the neighborhood.

Essentially, correntropy is a second order statistical measure (i.e. correlation) in kernel space, which corresponds to a non-second order measure in original space. Similarly, one can define other second order statistical measures, such as MSE, in kernel space. The MSE in kernel space is also called the *correntropic loss* (C-Loss) [19], [20]. It can be shown that minimizing the C-Loss is equivalent to maximizing the correntropy. In this paper, we define a non-second order measure in kernel space, called *kernel mean p-power error* (KMPE), which is the MPE in kernel space and, of course, is also a non-second order measure in original space. The KMPE will reduce to the C-Loss as $p = 2$, but with a proper p value can outperform the C-Loss when used as a loss function in robust learning. In the present work, we focus mainly on two application examples, *extreme learning machine* (ELM) [21], [22] and PCA. The ELM is a single-hidden-layer feedforward neural network (SLFN) with randomly generated hidden nodes, which can be used for regression, classification and many other learning tasks [21], [22]. The proposed KMPE will be used to develop robust ELM and PCA algorithms.

The rest of the paper is structured as follows. In section II, we define the KMPE, and give some basic properties. In section III, we apply the KMPE to ELM and PCA, and develop the ELM-KMPE and PCA-KMPE algorithms. In section IV, we present experimental results to demonstrate the desirable performance of the new algorithms. Finally in section V, we give the conclusion.

This work was supported by 973 Program (No. 2015CB351703) and National NSF of China (No. 91648208, No. 61372152).

Badong Chen, Lei Xing, Xin Wang and Nanning Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, 710049, China.(chenbd; nnzheng @mail.xjtu.edu.cn; xl2010; wangxin0420@stu.xjtu.edu.cn).

Jing Qin is with the Center of Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hongkong, China.(harryqinjingcn@gmail.com).

II. KERNEL MEAN P-POWER ERROR

A. Definition

Non-second order statistical measures can be defined elegantly as a second order measure in kernel space. For example, the correntropy between two random variables X and Y, is a correlation measure in kernel space, given by [10]

$$\begin{aligned} V(X,Y) &= \mathbf{E}\left[\langle \Phi(X), \Phi(Y)\rangle_{\mathcal{H}}\right] \\ &= \int \langle \Phi(x), \Phi(y)\rangle_{\mathcal{H}} dF_{XY}(x,y) \end{aligned} \tag{1}$$

where $\mathbf{E}[.]$ denotes the expectation operator, $F_{XY}(x,y)$ stands for the joint distribution function, and $\Phi(x) = \kappa(x,.)$ is a nonlinear mapping induced by a Mercer kernel $\kappa(.,.)$, which transforms x from the original space to a functional Hilbert space (or kernel space) $\mathcal{H}$ equipped with an inner product $\langle .,. \rangle_{\mathcal{H}}$ satisfying $\langle \Phi(x), \Phi(y)\rangle_{\mathcal{H}} = \kappa(x,y)$. Obviously, we have $V(X,Y) = \mathbf{E}\left[\kappa(X,Y)\right]$. In this paper, without mentioned otherwise, the kernel function is a Gaussian kernel, given by

$$\kappa(x,y) = \kappa_\sigma(x-y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \tag{2}$$

with σ being the kernel bandwidth. Similarly, the C-Loss as MSE in kernel space, can be defined by [14]

$$\begin{aligned} C(X,Y) &= \frac{1}{2}\mathbf{E}\left[\|\Phi(X) - \Phi(Y)\|_{\mathcal{H}}^2\right] \\ &= \frac{1}{2}\mathbf{E}\left[\langle \Phi(X) - \Phi(Y), \Phi(X) - \Phi(Y)\rangle_{\mathcal{H}}\right] \\ &= \frac{1}{2}\mathbf{E}\left[\langle \Phi(X), \Phi(X)\rangle_{\mathcal{H}} + \langle \Phi(Y), \Phi(Y)\rangle_{\mathcal{H}} - 2\langle \Phi(X), \Phi(Y)\rangle_{\mathcal{H}}\right] \\ &= \frac{1}{2}\mathbf{E}\left[2\kappa_\sigma(0) - 2\kappa_\sigma(X-Y)\right] \\ &= \mathbf{E}\left[1 - \kappa_\sigma(X-Y)\right] \end{aligned} \tag{3}$$

where $1/2$ is inserted to make the expression more convenient. It holds that $C(X,Y) = 1 - V(X,Y)$, hence minimizing the C-Loss will be equivalent to maximizing the correntropy. The *maximum correntropy criterion* (MCC) has drawn more and more attention recently due to its robustness to large outliers [10]–[18].

In this work, we define a new statistical measure in kernel space in a non-second order manner. Specifically, we generalize the C-Loss to the case of arbitrary power and define the mean p-power error (MPE) in kernel space, and call the new measure the kernel MPE (KMPE). Given two random variables X and Y, the KMPE is defined by

$$\begin{aligned} C_p(X,Y) &= 2^{-p/2}\mathbf{E}\left[\|\Phi(X) - \Phi(Y)\|_{\mathcal{H}}^p\right] \\ &= 2^{-p/2}\mathbf{E}\left[\left(\|\Phi(X) - \Phi(Y)\|_{\mathcal{H}}^2\right)^{p/2}\right] \\ &= 2^{-p/2}\mathbf{E}\left[\left(2 - 2\kappa_\sigma(X-Y)\right)^{p/2}\right] \\ &= \mathbf{E}\left[\left(1 - \kappa_\sigma(X-Y)\right)^{p/2}\right] \end{aligned} \tag{4}$$

where $p > 0$ is the power parameter. Clearly, the KMPE includes the C-Loss as a special case (when $p = 2$). In addition, given N samples $\{x_i, y_i\}_{i=1}^N$, the *empirical KMPE* can be easily obtained as

$$\hat{C}_p(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \kappa_\sigma(x_i - y_i)\right)^{p/2} \tag{5}$$

Since $\hat{C}_p(X,Y)$ is a function of the sample vectors $\mathbf{X} = [x_1, x_2, \cdots, x_N]^T$ and $\mathbf{Y} = [y_1, y_2, \cdots, y_N]^T$, one can also denote $\hat{C}_p(X,Y)$ by $\hat{C}_p(\mathbf{X},\mathbf{Y})$ if no confusion arises.

B. Properties

Some basic properties of the proposed KMPE are presented below.

Property 1: $C_p(X,Y)$ is symmetric, that is $C_p(X,Y) = C_p(Y,X)$.

Proof: Straightforward since $\kappa_\sigma(X-Y) = \kappa_\sigma(Y-X)$.

Property 2: $C_p(X,Y)$ is positive and bounded: $0 \le C_p(X,Y) < 1$, and it reaches its minimum if and only if $X = Y$.

Proof: Straightforward since $0 < \kappa_\sigma(X-Y) \le 1$, with $\kappa_\sigma(X-Y) = 1$ if and only if $X = Y$.

Property 3: As p is small enough, it holds that $C_p(X,Y) \approx 1 + \frac{p}{2}\mathbf{E}\left[\log\left(1 - \kappa_\sigma(X-Y)\right)\right]$.

Proof: The property holds since $\left(1 - \kappa_\sigma(X-Y)\right)^{p/2} \approx 1 + \frac{p}{2}\log\left(1 - \kappa_\sigma(X-Y)\right)$ for p small enough.

Property 4: As σ is large enough, it holds that $C_p(X,Y) \approx \left(2\sigma^2\right)^{-p/2}\mathbf{E}\left[|X-Y|^p\right]$.

Proof: Since $\exp(x) \approx 1 + x$ for x small enough, as $\sigma \to \infty$, we have

$$\begin{aligned} \left(1 - \kappa_\sigma(X-Y)\right)^{p/2} &= \left(1 - \exp\left(-\frac{(X-Y)^2}{2\sigma^2}\right)\right)^{p/2} \\ &\approx \left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2} \\ &= \left(2\sigma^2\right)^{-p/2}|X-Y|^p \end{aligned} \tag{6}$$

Remark: By Property 4, one can conclude that the KMPE will be, approximately, equivalent to the MPE when kernel bandwidth σ is large enough.

Property 5: Let $\boldsymbol{e} = \mathbf{X} - \mathbf{Y} = [e_1, e_2, \cdots, e_N]^T$, where $e_i = x_i - y_i$. if $p \ge 2$, the empirical KMPE $\hat{C}_p(\mathbf{X},\mathbf{Y})$ as a function of $\boldsymbol{e}$ is convex at any point satisfying $\|\boldsymbol{e}\|_\infty = \max_{i=1,2,\cdots,N}|e_i| \le \sigma$.

Proof: Since $\hat{C}_p(\mathbf{X},\mathbf{Y}) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \kappa_\sigma(e_i)\right)^{p/2}$, the Hessian matrix of $\hat{C}_p(\mathbf{X},\mathbf{Y})$ with respect to $\boldsymbol{e}$ is

$$H_{\hat{C}_p(\mathbf{X},\mathbf{Y})}(\boldsymbol{e}) = \left[\frac{\partial^2 \hat{C}_p(\mathbf{X},\mathbf{Y})}{\partial e_i \partial e_j}\right] = diag\left[\xi_1, \xi_2, \cdots, \xi_N\right] \tag{7}$$

where

$$\begin{aligned} \xi_i = &\frac{p}{4N\sigma^4}\left(1 - \kappa_\sigma(e_i)\right)^{(p-4)/2}\kappa_\sigma(e_i)\times \\ &\left\{(p-2)e_i^2\kappa_\sigma(e_i) - 2e_i^2\left(1 - \kappa_\sigma(e_i)\right) + 2\sigma^2\left(1 - \kappa_\sigma(e_i)\right)\right\} \end{aligned} \tag{8}$$

When $p \geq 2$, we have $\xi_i \geq 0$ if $|e_i| \leq \sigma$. Thus, for any point $\boldsymbol{e}$ with $\|\boldsymbol{e}\|_\infty \leq \sigma$, we have $H_{\hat{C}_p(\mathbf{X},\mathbf{Y})}(\boldsymbol{e}) \geq 0$.

Property 6: Given any point $\boldsymbol{e}$ with $\|\boldsymbol{e}\|_\infty > \sigma$, the empirical KMPE $\hat{C}_p(X,Y)$ will be convex at $\boldsymbol{e}$ if p is larger than a certain value.

Proof: From (8), if $|e_i| \leq \sigma$ and $p \geq 2$, or if $|e_i| > \sigma$ and $p \geq \frac{2\left[e_i^2-\sigma^2\right](1-\kappa_\sigma(e_i))}{e_i^2\kappa_\sigma(e_i)} + 2$, we have $\xi_i \geq 0$. So, it holds that $H_{\hat{C}_p(\mathbf{X},\mathbf{Y})}(\boldsymbol{e}) \geq 0$ if

$$p \geq \max_{\substack{i=1,\cdots,N \\ |e_i|>\sigma}} \left\{ \frac{2\left[e_i^2-\sigma^2\right](1-\kappa_\sigma(e_i))}{e_i^2\kappa_\sigma(e_i)} + 2 \right\} \tag{9}$$

This complete the proof.

Remark: According to Property 5 and 6, the empirical KMPE as a function of $\boldsymbol{e}$ is convex at any point with $\|\boldsymbol{e}\|_\infty \leq \sigma$. and it can also be convex at a point with $\|\boldsymbol{e}\|_\infty > \sigma$ if the power parameter p is larger than a certain value.

Property 7: Let $\mathbf{0}$ be an N-dimensional zero vector. Then as $\sigma \to \infty$ (or $x_i \to 0, i = 1, \cdots, N$), it holds that

$$\hat{C}_p(\mathbf{X},\mathbf{0}) \approx \frac{1}{N\left(\sqrt{2}\sigma\right)^p} \|\mathbf{X}\|_p^p \tag{10}$$

where $\|\mathbf{X}\|_p^p = \sum_{i=1}^{N} |x_i|^p$.

Proof: As σ is large enough, we have

$$\begin{aligned} \hat{C}_p(\mathbf{X},\mathbf{0}) &= \frac{1}{N}\sum_{i=1}^{N}\left(1-\kappa_\sigma(x_i)\right)^{p/2} \\ &\overset{(a)}{\approx} \frac{1}{N}\sum_{i=1}^{N}\left(1-\left(1-\frac{x_i^2}{2\sigma^2}\right)\right)^{p/2} \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i^2}{2\sigma^2}\right)^{p/2} \\ &= \frac{1}{N\left(\sqrt{2}\sigma\right)^p}\sum_{i=1}^{N}|x_i|^p \end{aligned} \tag{11}$$

Property 8: Assume that $|x_i| > \delta$, $\forall i : x_i \neq 0$, where δ is a small positive number. As $\sigma \to 0+$, minimizing the empirical KMPE $\hat{C}_p(\mathbf{X},\mathbf{0})$ will be, approximately, equivalent to minimizing the l_0-norm of $\mathbf{X}$, that is

$$\min_{\mathbf{X}\in\Omega} \hat{C}_p(\mathbf{X},\mathbf{0}) \sim \min_{\mathbf{X}\in\Omega} \|\mathbf{X}\|_0, \;\; as \;\; \sigma \to 0+ \tag{12}$$

where Ω denotes a feasible set of $\mathbf{X}$.

Proof: Let $\mathbf{X}_0$ be the solution obtained by minimizing $\|\mathbf{X}\|_0$ over Ω and $\mathbf{X}_C$ the solution achieved by minimizing $\hat{C}_p(\mathbf{X},\mathbf{0})$. Then $\hat{C}_p(\mathbf{X}_C,\mathbf{0}) \leq \hat{C}_p(\mathbf{X}_0,\mathbf{0})$, and

$$\begin{aligned} &\sum_{i=1}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_C)_i\right)\right)^{p/2}-1\right] \\ &\leq \sum_{i=1}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_0)_i\right)\right)^{p/2}-1\right] \end{aligned} \tag{13}$$

where $(\mathbf{X}_C)_i$ denotes the ith component of $\mathbf{X}_C$. It follows that

$$\begin{aligned} &\|\mathbf{X}_C\|_0 - N + \sum_{i=1,(\mathbf{X}_C)_i\neq 0}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_C)_i\right)\right)^{p/2}-1\right] \\ &\leq \|\mathbf{X}_0\|_0 - N + \sum_{i=1,(\mathbf{X}_0)_i\neq 0}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_0)_i\right)\right)^{p/2}-1\right] \end{aligned} \tag{14}$$

Hence

$$\begin{aligned} \|\mathbf{X}_C\|_0 - \|\mathbf{X}_0\|_0 \leq &\sum_{i=1,(\mathbf{X}_0)_i\neq 0}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_0)_i\right)\right)^{p/2}-1\right] \\ &- \sum_{i=1,(\mathbf{X}_C)_i\neq 0}^{N}\left[\left(1-\kappa_\sigma\left((\mathbf{X}_C)_i\right)\right)^{p/2}-1\right] \end{aligned} \tag{15}$$

Since $|x_i| > \delta$, $\forall i : x_i \neq 0$, as $\sigma \to 0+$ the right hand side of (15) will approach zero. Thus, if σ is small enough, it holds that

$$\|\mathbf{X}_0\|_0 \leq \|\mathbf{X}_C\|_0 \leq \|\mathbf{X}_0\|_0 + \varepsilon \tag{16}$$

where ε is a small positive number arbitrarily close to zero. This completes the proof.

Remark: From Property 7 and 8, one can see that the empirical KMPE $\hat{C}_p(\mathbf{X},\mathbf{0})$ behaves like an L_p norm of $\mathbf{X}$ when kernel bandwidth σ is very large, and like an L_0 norm of $\mathbf{X}$ when σ is very small.

III. APPLICATION EXAMPLES

There are many applications in areas of machine learning and signal processing that can employ the KMPE to solve robustly the relevant problems. In this section, we present two examples to investigate the benefits from the KMPE.

A. Extreme Learning Machine

The first example is about the Extreme Learning Machine (ELM), a single-hidden-layer feedforward neural network (SLFN) with random hidden nodes [21], [22]. With a quadratic loss function, the ELM usually requires no iterative tuning and the global optima can be solved in a batch mode. In the following, we use the KMPE as the loss function for ELM, and develop a robust algorithm to train the model. Since there is no closed-form solution under the KMPE loss, the new algorithm will be a fixed-point iterative algorithm.

Given N distinct training samples $\{\mathbf{x}_i, t_i\}_{i=1}^N$, with $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{id}]^T \in \mathbb{R}^d$ being the input vector and the $t_i \in \mathbb{R}$ target response, the output of a standard SLFN with L hidden nodes will be

$$y_i = \sum_{j=1}^{L} \beta_j f(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) \tag{17}$$

where $f(.)$ is an activation function, $\mathbf{w}_j = [w_{j1}, w_{j2}, ..., w_{jd}] \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ $(i = 1, 2, ..., L)$ are the learning parameters of the ith hidden node, $\mathbf{w}_j \cdot \mathbf{x}_i$ denotes the inner product of $\mathbf{w}_j$ and $\mathbf{x}_i$, and $\beta_j \in \mathbb{R}$ represents the weight parameter of the link connecting the jth hidden node to the output node. The above equation can be written in a vector form as

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta} \tag{18}$$

where $\mathbf{Y} = (y_1, ..., y_N)^T$, $\boldsymbol{\beta} = (\beta_1, ..., \beta_L)^T$ and

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1), & ... & f(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1), & ... & f(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{pmatrix} \quad (19)$$

represents the output matrix of the hidden layer. In general, the output weight vector $\boldsymbol{\beta}$ can be solved by minimizing the regularized MSE (or least squares) loss:

$$J_{MSE}(\boldsymbol{\beta}) = \sum_{i=1}^{N} e_i^2 + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 = \left\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\right\|_2^2 + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 \quad (20)$$

where $e_i = t_i - y_i$ is the error between the ith target response and the ith actual output, $\lambda \geq 0$ stands for the regularization parameter to prevent overfitting, and $\mathbf{T} = (t_1, ..., t_N)^T$ is the target response vector. With a pseudo inversion operation, one can easily obtain a unique solution under the loss (20), that is

$$\boldsymbol{\beta} = [\mathbf{H^T H} + \lambda \mathbf{I}]^{-1} \mathbf{H}^T \mathbf{T} \quad (21)$$

In order to obtain a solution that is robust with respect to large outliers, now we consider the following KMPE based loss function:

$$\begin{aligned} J_{KMPE}(\boldsymbol{\beta}) &= \hat{C}_p(\mathbf{T}, \mathbf{H}\boldsymbol{\beta}) + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 \\ &= \frac{1}{N} \sum_{i=1}^{N} \left(1 - \kappa_\sigma(e_i)\right)^{p/2} + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 \\ &= \frac{1}{N} \sum_{i=1}^{N} \left(1 - \exp(-\frac{e_i^2}{2\sigma^2})\right)^{p/2} + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 \end{aligned} \quad (22)$$

Note that different from the loss function in (20), the new loss function will be little influenced by large errors since the term $(1 - \kappa_\sigma(e_i))^{p/2}$ is upper bounded by 1.0.

Let $\frac{\partial}{\partial \boldsymbol{\beta}} J_{KMPE}(\boldsymbol{\beta}) = 0$. Then we derive

$$\begin{aligned} &\frac{\partial J_{KMPE}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \\ &\Rightarrow \frac{1}{N} \sum_{i=1}^{N} \left[\frac{-p}{2\sigma^2} (1 - \kappa_\sigma(e_i))^{(p-2)/2} \kappa_\sigma(e_i) e_i \boldsymbol{h}_i^T\right] + 2\lambda\boldsymbol{\beta} = 0 \\ &\Rightarrow \sum_{i=1}^{N} \left[-(1 - \kappa_\sigma(e_i))^{(p-2)/2} \kappa_\sigma(e_i) e_i \boldsymbol{h}_i^T\right] + \frac{4\sigma^2 N\lambda}{p}\boldsymbol{\beta} = 0 \\ &\Rightarrow \sum_{i=1}^{N} \left(\varphi(e_i) \boldsymbol{h}_i^T \boldsymbol{h}_i \boldsymbol{\beta} - \varphi(e_i) t_i \boldsymbol{h}_i^T\right) + \lambda'\boldsymbol{\beta} = 0 \\ &\Rightarrow \sum_{i=1}^{N} \left(\varphi(e_i) \boldsymbol{h}_i{}^T \boldsymbol{h}_i \boldsymbol{\beta}\right) + \lambda'\boldsymbol{\beta} = \sum_{i=1}^{N} \varphi(e_i) t_i \boldsymbol{h}_i^T \\ &\Rightarrow \boldsymbol{\beta} = [\mathbf{H}^T \boldsymbol{\Lambda} \mathbf{H} + \lambda' \mathbf{I}]^{-1} \mathbf{H}^T \boldsymbol{\Lambda} \mathbf{T} \end{aligned} \quad (23)$$

where $\lambda' = \frac{4\sigma^2 N}{p}\lambda$ is the ith row of $\mathbf{H}$, $\varphi(e_i) = (1 - \kappa_\sigma(e_i))^{(p-2)/2} \kappa_\sigma(e_i)$, and $\boldsymbol{\Lambda}$ is a diagonal matrix with diagonal elements $\boldsymbol{\Lambda}_{ii} = \varphi(e_i)$.

The derived optimal solution $\boldsymbol{\beta} = [\mathbf{H}^T \boldsymbol{\Lambda}\ \mathbf{H} + \lambda' \mathbf{I}]^{-1} \mathbf{H}^T \boldsymbol{\Lambda}\ \mathbf{T}$ is not a closed-form solution since the matrix $\boldsymbol{\Lambda}$ on the right-hand side depends on the weight vector $\boldsymbol{\beta}$ through $e_i = t_i - \boldsymbol{h}_i \boldsymbol{\beta}$. So it is actually a fixed-point equation. The true optimal solution can thus be solved by a fixed-point iterative algorithm, as summarized in Algorithm 1. This algorithm is referred to as the ELM-KMPE in this work.

Algorithm 1 ELM-KMPE

Input: samples $\{\mathbf{x}_i, t_i\}_{i=1}^N$
Output: weight vector $\boldsymbol{\beta}$
Parameters setting:number of hidden nodes L, regularization parameter λ', maximum iteration number M, kernel width σ, power parameter p and termination tolerance ε
Initialization: Set $\boldsymbol{\beta_0}$=0 and randomly initialize the parameters $\mathbf{w}_j$ and b_j ($j = 1, ..., L$)
1: **for** $k = 1, 2, ..., M$ **do**
2: Compute the error based on $\boldsymbol{\beta}_{k-1}$: $e_i = t_i - \boldsymbol{h}_i \boldsymbol{\beta}_{k-1}$
3: Compute the diagonal matrix $\boldsymbol{\Lambda}$: $\boldsymbol{\Lambda}_{ii} = \varphi(e_i)$
4: Update the weight vector$\boldsymbol{\beta}$: $\boldsymbol{\beta}_k = [\mathbf{H}^T \boldsymbol{\Lambda} \mathbf{H} + \lambda' \mathbf{I}]^{-1} \mathbf{H}^T \boldsymbol{\Lambda}\ \mathbf{T}$
5: **Until** $\left|J_{KMPE}(\boldsymbol{\beta}_k) - J_{KMPE}(\boldsymbol{\beta}_{k-1})\right| < \varepsilon$
6: **end for**

B. Principal Component Analysis

The second example is the Principal Component Analysis (PCA), one of the most popular dimensionality reduction methods [2]. Below we use the proposed KMPE as the loss function to derive a robust PCA algorithm.

Consider a set of samples $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, with d being the dimension number and n the sample number. The PCA methods try to find a projection matrix $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ to define a new orthogonal coordinate system that can optimally describe the variability in the data set. In L2-PCA, the projection matrix is solved by minimizing the following loss function [2]:

$$\begin{aligned} \ell_{L2}(\mathbf{W}) &= \left\|\tilde{\mathbf{X}} - \mathbf{W}\mathbf{V}\right\|_2^2 = \sum_{i=1}^{n} \left\|\mathbf{x}_i - \boldsymbol{\mu} - \sum_{k=1}^{m} \mathbf{w}_k v_{ki}\right\|_2^2 \\ &= \sum_{i=1}^{n} \sum_{j=1}^{d} \left(x_{ji} - \mu_j - \sum_{k=1}^{m} w_{jk} v_{ki}\right)^2 \end{aligned} \quad (24)$$

where $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_\mathbf{1}, ..., \tilde{\mathbf{x}}_\mathbf{n}]$ denotes the column-wise-zero-mean version of $\mathbf{X}$, with $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} - \boldsymbol{\mu}$, $\boldsymbol{\mu}$ is the sample mean of column vectors, and $\mathbf{V} = \mathbf{W}^T \tilde{\mathbf{X}} = [\mathbf{v}_1, ..., \mathbf{v}_n] \in \mathbb{R}^{m \times n}$ contains the principal components that are projected under the projection matrix $\mathbf{W}$.

In order to prevent the outliers in the edge data from corrupting the results of dimensionality reduction, we minimize the following robust cost function for PCA:

$$\begin{aligned} \ell_{KMPE}(\mathbf{W}, \boldsymbol{\mu}) &= \hat{C}_p\left(\tilde{\mathbf{X}}, \mathbf{W}\mathbf{W}^T \tilde{\mathbf{X}}\right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(1 - \kappa_\sigma\left(\mathbf{x}_i - \boldsymbol{\mu} - \mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu})\right)\right)^{p/2} \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(1 - \exp\left(-\frac{\left\|\mathbf{e}_i\right\|_2^2}{2\sigma^2}\right)\right)^{p/2} \\ &= \frac{1}{n} \sum_{i=1}^{n} \rho\left(\left\|\mathbf{e}_i\right\|_2\right) \end{aligned} \quad (25)$$

where $\mathbf{e}_i = \mathbf{x}_i - \boldsymbol{\mu} - \mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu})$. Indeed, the cost function $\rho\left(\left\|\mathbf{e}_i\right\|_2\right) = \left(1 - \exp\left(-\frac{\left\|\mathbf{e}_i\right\|_2^2}{2\sigma^2}\right)\right)^{p/2}$ belongs to the

M-estimation robust cost functions [23], [24], and minimizing the cost (25) is an M-estimation problem. It is instructive and useful to transform the minimization of (25) into a weighted least squares problem, which can be solved by iteratively reweighted least squares (IRLS). This method is originally proposed in [25] and successfully used in robust statistics [26], computer vision [27], [28], face recognition [29], [30] and PCA [31]. Here, the weighting matrix $\mathbf{\Lambda}$ is a diagonal matrix with elements $\Lambda_{ii} = \psi\left(\|\mathbf{e}_i\|_2\right)/\|\mathbf{e}_i\|_2$, where $\psi\left(\|\mathbf{e}_i\|_2\right) = \frac{\partial\rho\left(\|\mathbf{e}_i\|_2\right)}{\partial\|\mathbf{e}_i\|_2}$. In this way, the cost function (25) will be equivalent to the following weighted least squares cost:

$$\begin{aligned}&\tilde{\ell}_{KMPE}\left(\mathbf{W},\boldsymbol{\mu}\right)\\&=\sum_{i=1}^{n}\left(\mathbf{x}_i-\boldsymbol{\mu}-\mathbf{W}\mathbf{W}^T\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\right)^T\Lambda_{ii}\left(\mathbf{x}_i-\boldsymbol{\mu}-\mathbf{W}\mathbf{W}^T\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\right)\end{aligned} \tag{26}$$

where

$$\Lambda_{ii}=\left(1-\exp\left(-\frac{\|\mathbf{e}_i\|_2^2}{2\sigma^2}\right)\right)^{(p-2)/2}\exp\left(-\frac{\|\mathbf{e}_i\|_2^2}{2\sigma^2}\right) \tag{27}$$

Setting $\frac{\partial}{\partial\boldsymbol{\mu}}\tilde{\ell}_{KMPE}\left(\mathbf{W},\boldsymbol{\mu}\right)=\mathbf{0}$, we derive

$$\boldsymbol{\mu}=\sum_{i=1}^{n}\Lambda_{ii}\mathbf{x}_i\Big/\sum_{i=1}^{n}\Lambda_{ii} \tag{28}$$

In addition, we can easily obtain the following solution

$$\mathbf{W}=\arg\max_{\mathbf{W}}Tr\left(\mathbf{W}^T\tilde{\mathbf{X}}\mathbf{\Lambda}\tilde{\mathbf{X}}^T\mathbf{W}\right) \tag{29}$$

The optimization problem (29) is a weighted PCA that can be computed by solving the corresponding eigenvalue problem. The solution of (25) can thus be obtained by iterating (27), (28) and (29). This algorithm is called in this work the PCA-KMPE, which when $p = 2.0$ will perform the HQ-PCA [12]. To learn an m-dimensional subspace, one can use a trick as in [12] to learn a small m_r dimensional subspace to further eliminate the influence by outliers. The proposed PCA-KMPE is summarized in Algorithm 2.

Algorithm 2 PCA-KMPE

Input: input data $\mathbf{X}$
Output: projection matrix $\mathbf{W} \in R^{d\times m}$
Parameters setting: maximum iteration number M, kernel width σ, power parameter p and termination tolerance ε
Initialization: $\boldsymbol{\mu}_0 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$, $\mathbf{W}_0 = \mathbf{W}_{PCA}$(solution of the original PCA)
1: **for** $k = 1, 2, ..., M$ **do**
2: Compute the errors based on $\mathbf{W}_{k-1}$ and $\boldsymbol{\mu}_{k-1}$:
$\mathbf{e}_i = \left(\mathbf{x}_i - \boldsymbol{\mu}_{k-1}\right) - \mathbf{W}_{k-1}\mathbf{W}_{k-1}^T\left(\mathbf{x}_i - \boldsymbol{\mu}_{k-1}\right)$
3: Compute the diagonal matrix $\mathbf{\Lambda}$ using (27)
4: Update the sample mean using (28)
5: Update the projection matrix $\mathbf{w}_k$ by solving the eigenvalue problem (29)
6: **Until** $\|\mathbf{W}_{k-1} - \mathbf{W}_k\|_2 \le \varepsilon$
7: **end for**

The kernel width σ is an important parameter in PCA-KMPE. In general, one can employ the Silvermans rule [32], to adjust the kernel width:

$$\sigma^2 = 1.06\times\min\left\{\sigma_E,\frac{R}{1.354}\right\}\times(n)^{-1/5} \tag{30}$$

where σ_E is the standard deviation of $\|\mathbf{e}_i\|_2^2$ and R is the interquartile range.

TABLE II
TESTING RMSES OF FOUR ALGORITHMS

	ELM	RELM	ELM-RCC	ELM-KMPE
Uniform	0.5117	0.2234	0.1671	**0.1079**
Sine wave	0.3340	0.2498	0.2335	**0.1156**

IV. EXPERIMENTAL RESULTS

This section presents some experimental results to verify the advantages of the ELM-KMPE and PCA-KMPE developed in the previous section.

A. Function estimation with synthetic data

In this example, the sinc function estimation, a popular illustration example for nonlinear regression problem in the literature, is used to evaluate the performance of the proposed ELM-KMPE and other ELM algorithms, such as ELM [21], RELM [33] and ELM-RCC [34]. The synthetic data are generated by $y(i) = k\cdot\mathrm{sinc}\left(x(i)\right) + v(i)$, where $k = 8$,

$$sinc(x)=\begin{cases}\sin(x)/x & x\neq 0\\ 1 & x=0\end{cases} \tag{31}$$

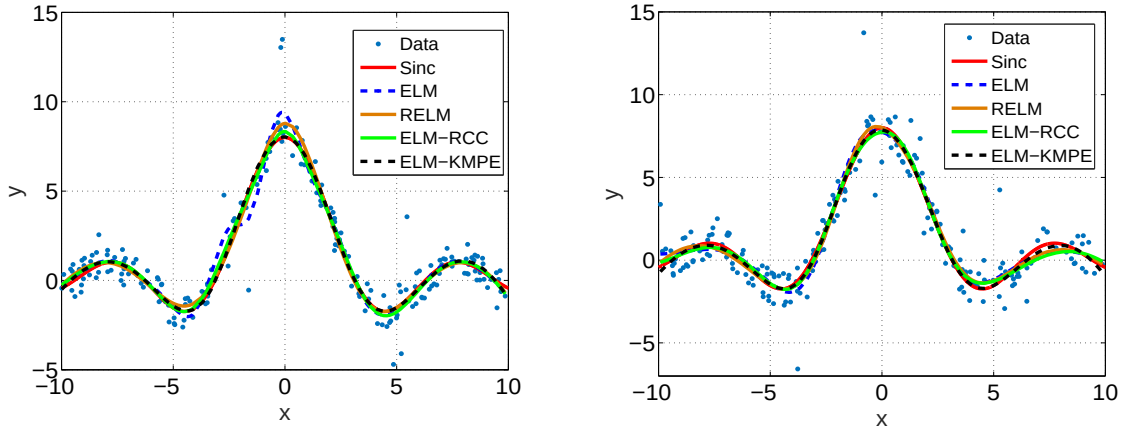
and $v(i)$ is a noise modeled as $v(i) = (1 - a(i))A(i) + a(i)B(i)$, where $a(i)$ is a binary iid process with probability mass $\Pr\{a(i) = 1\} = c$, $\Pr\{a(i) = 0\} = 1 - c$ ($0 \le c \le 1$), $A(i)$ denotes the background noise and $B(i)$ is another noise process to represent outliers. The noise processes $A(i)$ and $B(i)$ are mutually independent and both independent of $a(i)$. In this subsection, c is set at 0.1 and $B(i)$ is assumed to be a zero-mean Gaussian noise with variance 9.0. Two background noises are considered: a) Uniform distribution over $[-1.0, 1.0]$ and b) Sine wave noise $\sin(\omega)$, with ω uniformly distributed over $[0, 2\pi]$. In addition, the input data $x(i)$ are drawn uniformly from $[-10, 10]$. In the simulation, 200 samples are used for training and another 200 noise-free samples are used for testing. The RMSE is employed to measure the performance, calculated by

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\tilde{y}_i\right)^2} \tag{32}$$

where y_i and $\tilde{y}_i$ denote the target values and corresponding estimated values respectively, and N is the number of samples. The parameter settings of four algorithms under two distributions of $A(i)$ are summarized in Table 1, where L, λ (or λ'), σ and p denote the number of hidden layer nodes, regularization parameter, kernel width and the power parameter in ELM-KMPE. The estimation results and testing RMSEs are illustrated in Fig.1 and Table.2. It is evident that the ELM-KMPE achieves the best performance among the four algorithms.

B. Regression and classification on benchmark datasets

In this subsection, we compare the aforementioned four algorithms in regression and classification problems with



(a) (b)

Fig. 1. Sinc function estimation results with different background noises: (a) Uniform (b) Sine wave

TABLE I
PARAMETER SETTINGS OF FOUR ALGORITHMS UNDER TWO DISTRIBUTIONS OF $A(i)$

	ELM	RELM		ELM-RCC			ELM-KMPE			
	L	L	λ	L	λ	σ	L	λ'	σ	p
Uniform	20	90	5×10^{-5}	90	10^{-6}	1.5	90	2×10^{-6}	0.8	4
Sine wave	10	40	5×10^{-5}	25	5×10^{-6}	2	25	2.5×10^{-6}	1.2	3.4

benchmark datasets from UCI machine learning repository [35]. The details of the datasets are shown in Table 3 and 4. For each dataset, the training and testing samples are randomly selected form the set. In particular, the data for regression are normalized to the range $[0, 1]$. The parameter settings of the four algorithms for regression and classification experiments are presented in Table 5 and 6. For each algorithm, the parameters are experimentally chosen by fivefold cross-validation. The RMSE is used as the performance measure for regression. For classification, the performance is measured by the accuracy (ACC). Let p_i and t_i be the predicted and target labels of the ith sample. The ACC is defined by

$$ACC = \frac{1}{n}\sum_{i=1}^{n}\delta(t_i, map(p_i)) \tag{33}$$

where $\delta(x, y)$ is an indicator function, $\delta(x, y) = 1$ if $x = y$, otherwise $\delta(x, y) = 0$, and $map(\cdot)$ maps each predicted label to the equivalent target label. The Kuhn-Munkres algorithm [36] is employed to realize such a mapping. The "mean $\pm$ standard deviation" results of the RMSE and ACC during training and testing are shown in Table 7 and 8, where the best testing results are represented in bold for each data set. As one can see, in all the cases the proposed ELM-KMPE can outperform other algorithms.

C. Face reconstruction

In this part, we demonstrate the performance of the proposed PCA-KMPE algorithm by applying it to the face reconstruction task [37]. The Yale face database [38] is used, which contains 165 face image. Each image is normalized to

TABLE III
SPECIFICATION OF THE REGRESSION PROBLEM

Datasets	Features	Observations	
		Training	Testing
Servo	5	83	83
Concrete	9	515	515
Wine red	12	799	799
Housing	14	253	253
Airfoil	5	751	751
Slump	10	52	51
Yacht	6	154	154

TABLE IV
SPECIFICATION OF THE CLASSIFICATION PROBLEM

Datasets	Classes	Features	Observations	
			Training	Testing
Glass	7	11	114	100
Wine	3	13	89	89
Ecoli	8	7	180	156
User-Modeling	2	5	138	120
Wdbc	2	30	100	496
Leaf	36	14	180	160
Vehicle	4	18	500	346
Seed	3	7	110	100

TABLE V
PARAMETER SETTINGS OF FOUR ALGORITHMS IN REGRESSION

Datasets	ELM	RELM		ELM-RCC			ELM-KMPE			
	L	L	λ	L	λ	σ	L	λ'	σ	p
Servo	25	90	0.00001	65	0.8	0.0001	75	0.9	0.00001	1.6
Concrete	120	185	0.0002	200	0.6	0.000005	200	0.7	0.00005	2.2
Wine red	15	15	0.000002	25	0.3	0.001	115	0.5	0.001	2.2
Housing	40	180	0.001	200	0.8	0.001	200	0.9	0.002	2.2
Airfoil	130	200	0.0002	150	0.4	0.0000001	195	1.2	0.0000001	2.4
Slump	195	190	0.000025	165	0.6	0.000001	190	0.4	0.000002	2.8
Yacht	90	185	0.000025	195	0.4	0.0000001	175	1	0.0000001	1.0

TABLE VI
PARAMETER SETTINGS OF FOUR ALGORITHMS IN CLASSIFICATION

Datasets	ELM	RELM		ELM-RCC			ELM-KMPE			
	L	L	λ	L	λ	σ	L	λ'	σ	p
Glass	105	195	0.00005	185	1.5	0.001	180	1.4	0.001	2.8
Wine	15	15	0.000025	20	1.4	0.001	25	0.8	0.001	2.8
Ecoli	10	90	0.001	155	1.8	0.001	25	1.4	0.00005	2.8
User- Modeling	40	145	0.000025	125	1.8	0.000005	70	0.9	0.000002	2.4
Wdbc	40	145	0.000025	125	1.5	0.00005	50	1.2	0.00005	2.2
Leaf	70	130	0.00001	200	1.7	0.000025	180	1.3	0.0001	2.8
Vehicle	130	155	0.00001	195	1.3	0.00001	200	1	0.00005	2.4
Seed	30	130	0.0001	200	1.5	0.001	170	1.5	0.001	2.8

TABLE VII
PERFORMANCE COMPARISON OF FOUR ALGORITHMS WITH BENCHMARK REGRESSION DATASETS

Datasets	ELM		RELM		ELM-RCC		ELM-KMPE	
	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE
Servo	0.0741±0.0126	0.1183±0.0204	0.0720±0.0106	0.1036±0.0152	0.0739±0.0106	0.1032±0.0148	0.0570±0.0108	**0.1022±0.0184**
Concrete	0.0612±0.0026	0.0994±0.0013	0.0742±0.0025	0.0914±0.0042	0.0559±0.0018	0.0879±0.0077	0.0577±0.0021	**0.0864±0.0058**
Wine red	0.1280±0.0031	0.1312±0.0032	0.1282±0.0031	0.1309±0.0031	0.1264±0.0031	0.1306±0.0032	0.1198±0.0028	**0.1302±0.0035**
Housing	0.0728±0.0070	0.0994±0.0120	0.0502±0.0044	0.0835±0.0100	0.0493±0.0046	0.0832±0.0099	0.0554±0.0045	**0.0821±0.0101**
Airfoil	0.0664±0.0027	0.0942±0.0099	0.0967±0.0061	0.1025±0.0058	0.0742±0.0026	0.0896±0.0050	0.0695±0.0029	**0.0880±0.0058**
Slump	0±0	0.0429±0.0091	0.0066±0.0041	0.0424±0.0097	0.0001±0	0.0423±0.0130	0.0028±0.0004	**0.0410±0.0107**
Yacht	0.0040±0.0004	0.0740±0.1267	0.0370±0.0079	0.0530±0.0086	0.0126±0.0008	0.0333±0.0086	0.0051±0.0009	**0.0250±0.0147**

64×64 pixels, and the values of the pixels are set in $[0, 255]$. In our experiment, two types of outliers are considered. For the first type, some images are randomly selected, and the selected images are occluded by a rectangular area, where pixels are randomly set at either 0 or 255, and the location of the rectangular area is randomly determined. For the second type, all pixels of the selected images are set at either 0 or 255. Some examples of the first type are illustrated in Fig.2. The reconstruction performance is measured by the *average reconstruction error*, defined by [37]

$$e(m) = \frac{1}{n}\sum_{i=1}^{n} \left\| (\mathbf{x}_i^{org} - \boldsymbol{\mu}) - \mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu}) \right\|_2 \qquad (34)$$

where $\mathbf{x}_i^{org}$ and $\mathbf{x}_i$ denote, respectively, the original unoccluded image and corresponding training image. For comparison purpose, we also demonstrate the performance of the PCA [2], PCA-L1 [37], R1-PCA [39], PCA-GM R1-PCA [40] and HQ-PCA [12]. In the experiment, the kernel widths of the PCA-KMPE and HQ-PCA are selected by (30). The parameter of HQ-PCA, PCA-GM and PCA-KMPE is set at 10. The average

TABLE VIII
PERFORMANCE COMPARISON OF FOUR ALGORITHMS WITH BENCHMARK CLASSIFICATION DATASETS

Datasets	ELM		RELM		ELM-RCC		ELM-KMPE	
	Training ACC	Testing ACC	Training ACC	Testing ACC	Training ACC	Testing ACC	Training ACC	Testing ACC
Glass	95.32±3.39	77.62±9.55	96.04±1.72	92.50±4.31	96.80±2.71	93.24±3.38	97.85±2.03	**94.54±3.12**
Wine	99.55±0.70	96.91±2.07	99.65±0.67	96.92±2.07	99.85±0.44	97.43±1.95	99.91±0.35	**97.58±1.73**
Ecoli	90.45±3.46	80.65±3.51	92.61±3.44	82.19±3.11	92.48±3.24	82.27±2.93	93.50±3.39	**82.35±2.77**
User-Modeling	92.80±1.99	84.17±3.63	93.47±1.68	85.47±3.26	94.02±1.63	85.57±3.42	93.17±1.82	**86.29±3.08**
Wdbc	93.07±2.11	84.81±3.43	93.52±2.07	85.86±3.31	92.34±2.16	86.63±3.27	91.01±2.02	**87.09±3.20**
Leaf	93.63±1.28	68.86±3.92	95.91±1.15	71.51±3.71	95.01±1.53	71.56±3.88	95.21±1.48	**73.87±4.20**
Vehicle	92.87±1.03	81.14±1.91	94.01±0.97	81.51±1.99	94.88±0.80	81.61±2.03	95.80±0.79	**82.23±2.25**
Seed	98.48±1.07	92.26±2.65	98.75±0.93	94.40±2.13	98.30±1.03	94.65±1.95	98.65±1.05	**95.01±1.96**

Fig. 2. Some examples of the original images (upper row) and corresponding contaminated images (low row)

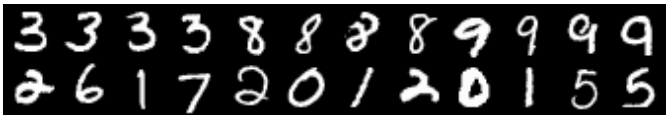


Fig. 7. Selected digital images from MNIST handwritten database

reconstruction errors of the six PCA algorithms versus the number of principal components under two types of outliers are illustrated in Fig.3 and 4. Evidently, the PCA-KMPE algorithm achieves the best performance among all the tested methods.

The effectiveness of the proposed PCA-KMPE can also be verified by visualizing the eigenfaces and reconstructed images. The eigenfaces obtained by PCA, R1-PCA, PCA-L1, HQ-PCA, PCA-GM and PCA-KMPE are shown in Fig.5. Due to space limitation, for each method only ten eigenfaces are presented, with $m = 10$. In addition, Fig.6 shows the face reconstruction results. These results are achieved under the occlusion and dummy noises (the numbers of the inlier and outlier images are (150, 15)) with the number of the extracted features being 50. Since there are some noisy images (occlusion or dummy) in the training set, most of the eigenfaces (especially those obtained by PCA, R1-PCA and PCA-L1) are contaminated. However, the eigenfaces of PCA-KMPE look very good in visualization. From Fig.6, one can observe that PCA-KMPE can well eliminate the influence by outliers.

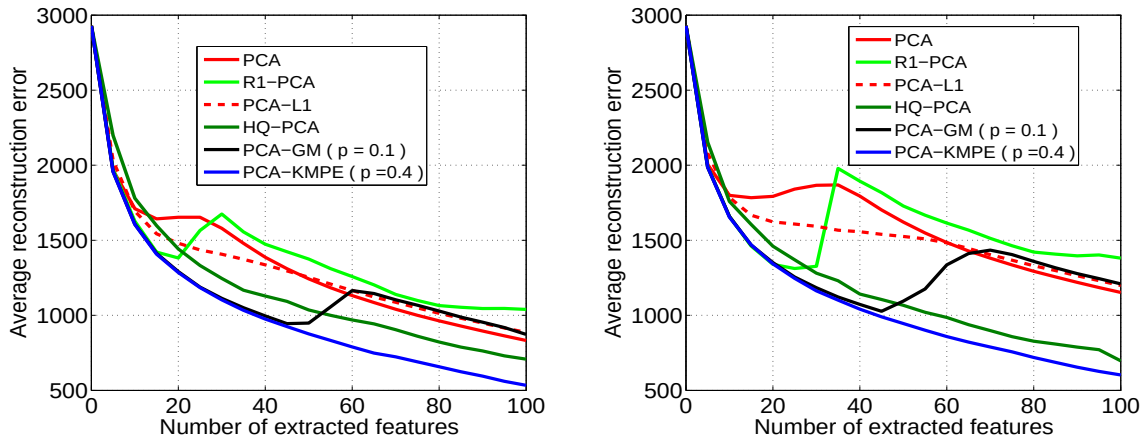
D. Clustering

Theoretical analysis and experimental results [12], [39]–[41] in the literature show that PCA methods can be used as a preprocessing step to improve the clustering accuracy of K-means. In the last part, we apply the proposed PCA-KMPE algorithm to a clustering problem with outliers. Two databases, MNIST handwritten digits database and Yale Face database, are chosen in our experiment. The MNIST handwritten digits data contain 60000 samples in training set and 10000 samples in testing set. In the experiment, we randomly select 300 samples of digits $\{3, 8, 9\}$ from the first 10000 samples in training set. Accordingly, 60 samples of other digits as outliers, are selected from the same 10000 samples. Thus the numbers of the outliers and inliers are 60 and 300, respectively. The selected samples are normalized to unit norm before experiment. In Fig.7, the upper and lower rows show the randomly selected inlier and outlier digits from the database. The second database, Yale Face, contains 165 grayscale images of 15 individuals, namely 15 classes. In the experiment, 15 dummy images contaminate the database. The goal is thus to learn a projection matrix from the training data (360 handwritten digital images or 180 faces images) using a PCA method, and obtain the testing results with testing data (noise free) on subspaces. Then we use the K-means algorithm to cluster the PCA results into 3 or 15 classes.

We use the clustering accuracy (ACC) and normalized mutual information (NMI) of K-means on subspaces, to quantitatively evaluate the performance of the aforementioned six PCA methods. Let $\boldsymbol{p}$ and $\boldsymbol{t}$ be the predicted and target label vectors, NMI is defined as

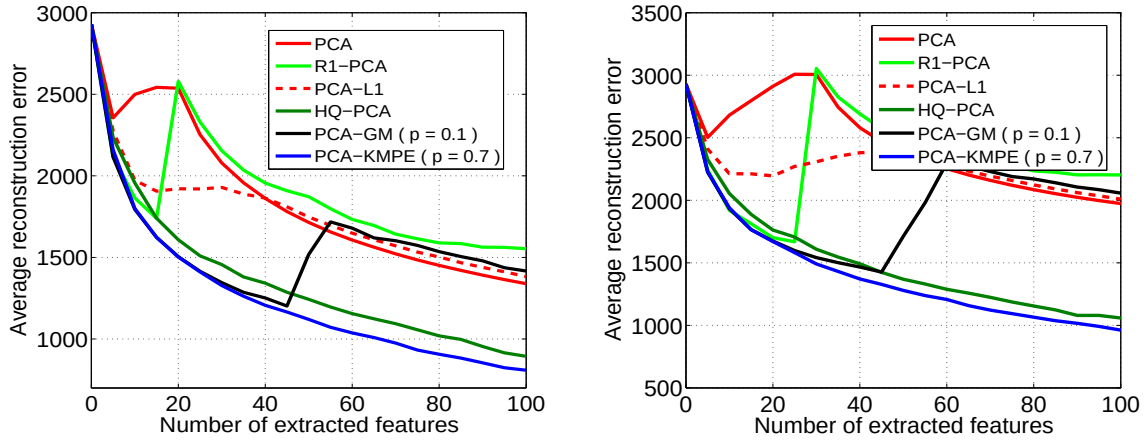
$$NMI = \frac{I(\boldsymbol{p}, \boldsymbol{t})}{\sqrt{H(\boldsymbol{p})H(\boldsymbol{t})}} \tag{35}$$

where $I(\boldsymbol{p}, \boldsymbol{t})$ is the mutual information between $\boldsymbol{p}$ and $\boldsymbol{t}$, and $H(\boldsymbol{p})$ and $H(\boldsymbol{t})$ are the entropies of $\boldsymbol{p}$ and $\boldsymbol{t}$. Clearly, the higher the values of ACC and NMI, the better the clustering performance. The clustering results on the two databases with different PCA methods are shown in Table 9 and 10, in which the best results under the same dimension number of subspaces are represented in bold. One can see that PCA-KMPE usually achieves the best performance among the six methods.



(a) (b)

Fig. 3. Average reconstruction errors of different PCA algorithms under occlusion images, where the numbers of inliers and outliers are: (a) (150,15); (b) (140,25).

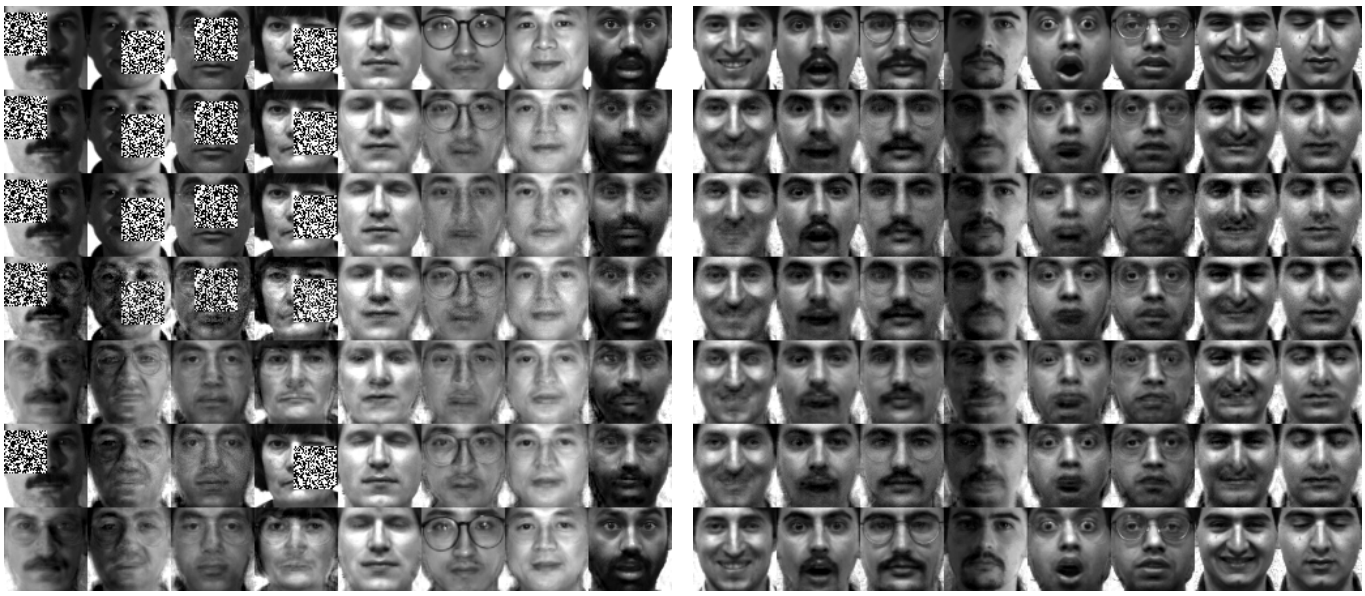


(a) (b)

Fig. 4. Average reconstruction errors of different PCA algorithms under dummy images, where the numbers of inliers and outliers are: (a) (150, 15); (b) (140, 25).

(a) (b)

Fig. 5. Eigenfaces of six PCA algorithms ($m = 10$). The images in each row are obtained by PCA, R1-PCA, PCA-L1, HQ-PCA, PCA-GM and PCA-KMPE. (a) occlusion noise; (b) dummy noise



(a) (b)

Fig. 6. Reconstructed images of six PCA algorithms. The first row shows the training images contaminated with (a) occlusion noise, (b) dummy noise. The rest rows show the images reconstructed by PCA, R1-PCA, PCA-L1, HQ-PCA, PCA-GM and PCA-KMPE.

TABLE IX
CLUSTERING ACCURACY (%) OF THE K-MEANS ON SUBSPACES OF THE DIGITAL IMAGES '3', '8' AND '9'

m	PCA	R1-PCA	L1-PCA	HQ-PCA	PCA-GM(p=0.3)	PCA-KMPE(p=10)
50	68.22±6.62	69.63±6.32	63.02±5.86	69.92±6.12	68.05±7.23	**71.83±6.06**
100	67.62±5.87	67.31±5.65	66.31±5.52	67.98±5.77	67.94±6.08	**69.53±6.61**
150	68.27±6.23	68.56±5.91	67.34±6.44	69.04±5.67	68.38±5.76	**69.55±5.71**
200	67.87±6.13	69.09±5.72	68.07±6.90	69.35±6.45	68.15±6.24	**69.91±6.85**
250	67.76±6.45	67.10±6.22	67.34±5.81	**68.62±6.25**	67.76±6.45	68.57±6.49
300	67.19±5.99	67.15±6.03	67.18±5.99	67.85±6.69	67.18±5.99	**68.03±6.55**

TABLE X
ACC AND NMI OF THE K-MEANS ON SUBSPACES OF YALE FACE DATABASE WITH DUMMY NOISE

m		PCA	R1-PCA	L1-PCA	HQ-PCA	PCA-GM(p=0.3)	PCA-KMPE(p=10)
20	ACC	0.4872±0.0433	0.4898±0.0382	0.4886±0.0380	0.4832±0.0410	0.4933±0.0419	**0.4933±0.0366**
	NMI	0.5618±0.0277	0.5567±0.0225	0.5620±0.0227	0.5375±0.0262	**0.5637±0.0256**	0.5604±0.0233
40	ACC	0.4809±0.0462	0.4848±0.0365	**0.4981±0.0405**	0.4907±0.0413	0.4859±0.0429	0.4927±0.0446
	NMI	0.5589±0.0317	0.5596±0.0254	0.5682±0.0264	0.5576±0.0275	0.5612±0.0270	**0.5715±0.0289**
60	ACC	0.4775±0.0444	0.4923±0.0498	0.4980±0.0400	0.4925±0.0390	0.4935±0.0398	**0.5133±0.0401**
	NMI	0.5570±0.0292	0.5657±0.0318	0.5693±0.0274	0.5617±0.0272	0.5658±0.0278	**0.5809±0.0281**
80	ACC	0.4938±0.0408	0.4887±0.0428	0.4903±0.0476	0.4900±0.0337	0.4851±0.0444	**0.5020±0.0432**
	NMI	0.5651±0.0291	0.5664±0.0295	0.5655±0.0330	0.5608±0.0268	0.5633±0.0319	**0.5741±0.0284**
100	ACC	0.4856±0.0430	0.4801±0.0422	0.4891±0.0489	0.4892±0.0423	0.4812±0.0497	**0.4956±0.0423**
	NMI	0.5648±0.0277	0.5580±0.0318	0.5640±0.0324	0.5587±0.0284	0.5602±0.0339	**0.5694±0.0292**

V. CONCLUSION

A new statistical measure in kernel space is proposed in this work, called the *kernel mean-p power error* (KMPE), which generalizes the *correntropic loss* (C-Loss) to the case of arbitrary power, and some basic properties are presented. In addition, we consider two application examples, *extreme learning machine* (ELM) and *principal component analysis* (PCA), and two robust learning algorithms are developed by using KMPE as loss function, namely ELM-KMPE and PCA-KMPE. Experimental results show that the new algorithms can consistently outperform some existing methods in function estimation, regression, classification, face reconstruction and clustering.

REFERENCES

[1] Jose C Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[2] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[3] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[4] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[5] Michael J Black, Guillermo Sapiro, David H Marimont, and David Heeger. Robust anisotropic diffusion. *IEEE Transactions on image processing*, 7(3):421–432, 1998.

[6] Rene K Boel, Matthew R James, and Ian R Petersen. Robustness and risk-sensitive filtering. *IEEE Transactions on Automatic Control*, 47(3):451–461, 2002.

[7] Soo-Chang Pei and Chien-Cheng Tseng. Least mean p-power error criterion for adaptive fir filter. *IEEE Journal on Selected Areas in Communications*, 12(9):1540–1547, 1994.

[8] Badong Chen, Lei Xing, Zongze Wu, Junli Liang, José C Príncipe, and Nanning Zheng. Smoothed least mean p-power error criterion for adaptive filtering. *Digital Signal Processing*, 40:154–163, 2015.

[9] Badong Chen, Yu Zhu, Jinchun Hu, and Jose C Principe. *System parameter identification: information criteria and algorithms*. Newnes, 2013.

[10] Weifeng Liu, Puskal P Pokharel, and José C Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11):5286–5298, 2007.

[11] Ran He, Tieniu Tan, and Liang Wang. Robust recovery of corrupted low-rankmatrix by implicit regularizers. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):770–783, 2014.

[12] Ran He, Bao-Gang Hu, Wei-Shi Zheng, and Xiang-Wei Kong. Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing*, 20(6):1485–1494, 2011.

[13] Badong Chen, Xi Liu, Haiquan Zhao, and Jose C Principe. Maximum correntropy kalman filter. *Automatica*, 76:70–77, 2017.

[14] Badong Chen, Lei Xing, Haiquan Zhao, Nanning Zheng, and Jose C Principe. Generalized correntropy for robust adaptive filtering. *IEEE Transactions on Signal Processing*, 64(13):3376–3387, 2016.

[15] Badong Chen, Jianji Wang, Haiquan Zhao, Nanning Zheng, and José C Príncipe. Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Processing Letters*, 22(10):1723–1727, 2015.

[16] Badong Chen, Lei Xing, Junli Liang, Nanning Zheng, and Jose C Principe. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE signal processing letters*, 21(7):880–884, 2014.

[17] Fotios D Mandanas and Constantine L Kotropoulos. Robust multidimensional scaling using a maximum correntropy criterion. *IEEE Transactions on Signal Processing*, 65(4):919–932, 2016.

[18] Fei Zhu, Abderrahim Halimi, Paul Honeine, Badong Chen, and Nanning Zheng. Correntropy maximization via admm-application to robust hyperspectral unmixing. *arXiv preprint arXiv:1602.01729*, 2016.

[19] Abhishek Singh, Rosha Pokharel, and Jose Principe. The c-loss function for pattern classification. *Pattern Recognition*, 47(1):441–453, 2014.

[20] Liangjun Chen, Hua Qu, Jihong Zhao, Badong Chen, and Jose C Principe. Efficient and robust deep learning with correntropy-induced loss function. *Neural Computing and Applications*, 27(4):1019–1031, 2016.

[21] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[22] Guang-Bin Huang, Dian Hui Wang, and Yuan Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.

[23] RARD Maronna, Douglas Martin, and Victor Yohai. *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.

[24] Peter J Huber. Wiley series in probability and mathematics statistics. *Robust Statistics*, pages 309–312, 1981.

[25] Albert E Beaton and John W Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[26] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[27] Shang-Hong Lai. Robust image matching under partial occlusion and spatially varying illumination change. *Computer Vision and Image Understanding*, 78(1):84–98, 2000.

[28] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.

[29] Chia-Po Wei and Yu-Chiang Frank Wang. Undersampled face recognition via robust auxiliary dictionary learning. *IEEE Transactions on Image Processing*, 24(6):1722–1734, 2015.

[30] Ran He, Wei-Shi Zheng, and Bao-Gang Hu. Maximum correntropy criterion for robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1561–1576, 2011.

[31] Fernando De La Torre and Michael J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[32] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[33] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

[34] Hong-Jie Xing and Xin-Mei Wang. Training extreme learning machine via regularized correntropy criterion. *Neural Computing and Applications*, 23(7-8):1977–1986, 2013.

[35] Andrew Frank and Arthur Asuncion. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of Information and Computer Science*, 213, 2010.

[36] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[37] Nojun Kwak. Principal component analysis based on l1-norm maximization. *IEEE transactions on pattern analysis and machine intelligence*, 30(9):1672–1680, 2008.

[38] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

[39] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006.

[40] Jiyong Oh and Nojun Kwak. Generalized mean for robust principal component analysis. *Pattern Recognition*, 54:116–127, 2016.

[41] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.