δ-Norm-Based Robust Regression With Applications to Image Analysis

Shuo Chen, Jian Yang, *Member, IEEE*, Yang Wei, Lei Luo, Gui-Fu Lu, and Chen Gong, *Member, IEEE*

***Abstract*—Up to now, various matrix norms (e.g., l_1-norm, l_2-norm, $l_{2,1}$-norm, etc.) have been widely leveraged to form the loss function of different regression models, and have played an important role in image analysis. However, the previous regression models adopting the existing norms are sensitive to outliers and, thus, often bring about unsatisfactory results on the heavily corrupted images. This is because their adopted norms for measuring the data residual can hardly suppress the negative influence of noisy data, which will probably mislead the regression process. To address this issue, this paper proposes a novel δ(delta)-norm to count the nonzero blocks around an element in a vector or matrix, which weakens the impacts of outliers and also takes the structure property of examples into account. After that, we present the δ-norm-based robust regression (DRR) in which the data examples are mapped to the kernel space and measured by the proposed δ-norm. By exploring an explicit kernel function, we show that DRR has a closed-form solution, which suggests that DRR can be efficiently solved. To further handle the influences from mixed noise, DRR is extended to a multiscale version. The experimental results on image classification and background modeling datasets validate the superiority of the proposed approach to the existing state-of-the-art robust regression models.**

***Index Terms*—Image analysis, kernel function, loss metric, outlier, robust regression (RR).**

I. Introduction

In COMPUTER vision and machine learning communities, l_p-norms, such as l_1-norm and l_2-norm, have been extensively employed to form the residual term of various regression models. For example, l_2-norm-based regression has been widely used in object detection [4], face alignment [61], feature selection [27], etc., while l_1-norm has become the core of a variety of vision tasks, such as robust face classification [44], [52] and feature extraction [43], which is considered to reduce the noise corruptions [6].

l_p-norms are defined as $\|\boldsymbol{x}\|_p = (\sum_{i=1}^{d} |x_i|^p)^{(1/p)}$, with $\boldsymbol{x} \in \mathbb{R}^d$ being a d-dimensional vector and $0 < p < \infty$. They are often taken as regularizers in many regression models. Wright *et al.* [46] presented a sparse representation classifier (SRC) which adopts l_1-norm to drive the solution vector as sparse as possible. Zhang *et al.* [55] investigated the contribution of representation coefficients to classification performance and found that the collaborative representation of coefficients is more meaningful than l_1-norm-based sparsity constraint. Therefore, they presented a collaborative representation classifier (CRC) incorporating l_2-norm. Besides, Cai *et al.* [5] investigated the probabilistic collaborative subspace for maximum-likelihood estimation (MLE), and derived the probabilistic CRC (ProCRC) model. l_p-norms also are widely utilized as the residual terms of regression models, due to their convexity for $p \geq 1$ cases. Lu *et al.* [29] used l_1-norm to fit the regression error for robust dictionary learning (DL). Nie *et al.* [33] introduced $l_{2,1}$-norm in their loss function to perform feature selection. Practically, the minimization of l_p-norms exactly satisfies MLE when the noise is independent and identically distributed (i.i.d.) [11].

However, the i.i.d. noise rarely occurs in image-related tasks as images are often corrupted by the abnormal noise (e.g., occlusions or illumination changes). The abnormal noise might not be i.i.d., and parts of components are arbitrarily corrupted. Data points are treated as outliers when they are corrupted by abnormal noises [58]. More important, the adverse impacts caused by outliers are also significantly amplified by the employed l_p-norms [28], because the intraclass variations corresponding to outliers are much larger than those induced by normal ones [50]. Consequently, existing l_p-norms-based regression approaches are not robust, and can hardly reconstruct the practical corrupted data. To enhance the robustness, some recent works introduced various reformed residual metrics instead of l_p-norms. Yang *et al.* [52] imposed a general probability density function on the regression residual, and presented the robust sparse coding (RSC). He *et al.* [11]

Manuscript received March 31, 2018; revised September 25, 2018; accepted February 6, 2019. This work was supported in part by the National Science Fund (NSF) of China under Grant U1713208, Grant 61472187, and Grant 61602246, in part by the Program for Changjiang Scholars, in part by the NSF of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the "Summit of the Six Top Talents" Program under Grant DZXX-027, in part by the "Innovative and Entrepreneurial Doctor" Program of Jiangsu Province, in part by the "Lift Program for Young Talents" of Jiangsu Province, in part by the "CAST Lift Program for Young Talents," in part by the "Innovation Program for Graduate Students" of Jiangsu Province, and in part by the "Outstanding Ph.D. of NUST" Program under Grant AE88902. This paper was recommended by Associate Editor G. C. Anagnostopoulos. *(Corresponding authors: Jian Yang; Chen Gong.)*

S. Chen, J. Yang, Y. Wei, L. Luo, and C. Gong are with the PCA Laboratory, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: shuochen@njust.edu.com; csjyang@njust.edu.cn; csywei@njust.edu.cn; zzdxpyy3001@163.com; chen.gong@njust.edu.cn).

G.-F. Lu is with the School of Computer Science and Information, Anhui Polytechnic University, Anhui 241000, China (e-mail: luguifu_jsj@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2901248

TABLE I
COMPARISON OF VARIOUS METRICS WITH DIFFERENT PROPERTIES

Metrics	**Global Optimization**	**Closed-Form Solution**	**Generalization of l_0-norm**	**Approximation way to l_0-norm**
l_1-norm [46]	✓	×	×	convex envelope
cos-loss [28]	✓	✓	×	constraint on upper bound
correntropy-induced-loss [49]	×	×		
capped-l_1-norm [15]				
weighted-l_1-norm [17]				
nuclear-norm [51]	✓	×	×	convex envelope on singular values
δ-norm (ours)	✓	✓	✓	**infinite approximation**

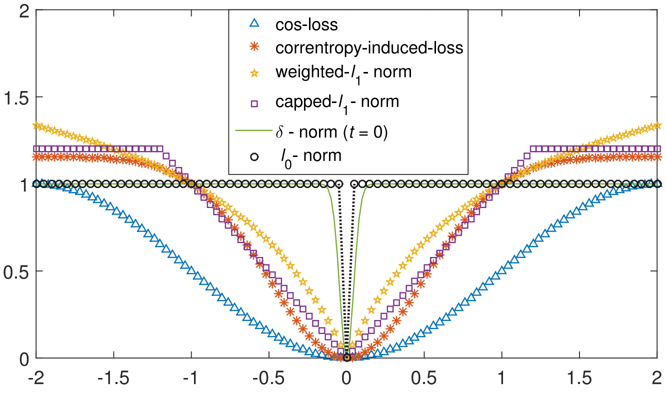


Fig. 1. Comparison of various robust metrics: cos-loss [28], correntropy-induced-loss [49], weighted-l_1-norm [17] , capped-l_1-norm [15], and δ-norm (with the special case $t = 0$ for approximating l_0-norm). The parameters in various metrics are set to their recommend values, while the proposed δ-norm is depicted with numerical approximation ($p = 100$) in the optimization model.

proposed the correntropy-based sparse representation (CESR), which uses a maximum correntropy criterion to characterize the regression residual. The Huber estimator was utilized by robust archetypal analysis (RAA) [7] to reduce the response of severe noise. Although bounded metrics are utilized in these models, RSC, CESR, and RAA are all probability-based methods, so they depend on several probability assumptions such as the asymptotically normally distributed data [13].

More recently, the l_0-norm (also called count-norm) is acknowledged as an effective metric for robust label fittings and data regression [12], [49]. However, the l_0-norm-based regression models are always NP-hard [1], [56], which cannot be solved directly. Therefore, some algebraic techniques are utilized to approximate l_0-norm. Liwicki *et al.* [28] proposed the Euler principle component analysis (Euler-PCA), which suggests using the cosine similarity metric in the kernel space to suppress severe noise. Kang *et al.* [17] introduced a weighted function of l_1-norm to approximate l_0-norm. Jiang *et al.* [15] developed the capped-l_1-norm to approximate the l_0-norm and evaluate the regression loss. Besides, the correntropy-induced-loss was validated to be a more precise approximation for l_0-norm [49]. To further approximate the l_0-norm, some works [10], [45] directly optimized l_0-norm loss function by means of proximal algorithms. These improvements are all based on vectors and, thus, ignore the structured information of noise. Toward this end, Yang *et al.* [51] utilized the rank function to characterize the structure of the noise image, and employed the nuclear norm [26] to achieve the low-rank effect. Interestingly, the rank function and nuclear norm can also be regarded as the generalized l_0-norm and l_1-norm, respectively, which are performed on the matrix singular values. In summary, the aforementioned methods only obtain rough approximations to l_0-norm (as shown in Fig. 1), so they are still not sufficiently robust. Moreover, since most robust metrics are nonconvex, their corresponding regressions cannot be solved by the global minimizers.

The target of a robust regression (RR) model is to drop the noise part and meanwhile preserve the clean information [13], [19]. However, as we reviewed above, existing regression models are not robust enough to severe noises, because their residual metrics are still sensitive to the noise values. Motivated by this standpoint, we construct a new distance metric called δ-norm[1] to fit the practical noise. The proposed δ-norm generalizes the l_0-norm by counting the number of nonzero neighbors rather than the nonzero feature elements. Table I shows the main differences between the δ-norm and the traditional robust metrics. From the table, we can find that the proposed δ-norm is more precise and general than other approximations to l_0-norm. As a generalized l_0-norm, the proposed δ-norm inherits the robustness of l_0-norm, and improves the fitting capability for image structures. This is because δ-norm counts the number of nonzero blocks in a data example, which means that the regression coefficients will be penalized if one element is not fitted and, thus, it forces the loss function to reconstruct the image block by block rather than pixel by pixel. With the robustness and structure attributes, the δ-norm is able to handle not only sparse noise (e.g., salt and pepper noise) but also structured noise. We subsequently present an RR model called δ-norm-based RR (DRR), by leveraging the proposed δ-norm in loss evaluation. Then, we explore a specific kernel function [48] to obtain the closed-form solution to DRR. The main contributions of this paper can be summarized as follows.

1) We propose a novel metric called δ-norm to generalize l_0-norm. The degenerated form of δ-norm is equivalent to l_0-norm and, thus, outperforms all existing approximations to l_0-norm. An RR model is implemented by employing δ-norm to characterize the regression residual.
2) We develop a type of general and efficient kernel trick to achieve the closed-form solution with global optimum to DRR. The proposed DRR model is further extended to a

[1] The proposed norm considers the neighborhood information within an example, so we use the neighborhood symbol δ [36] to name our proposed norm.

multiscale version called MDRR. Experiments validate the effectiveness of the proposed method.

We compare our method with various RR models with different loss functions. Moreover, the proposed δ-norm can also be applied to different regression models besides our adopted form.

The remainder of this paper is organized as follows. Section II defines the δ-norm in detail, and presents the DRR, including the model, solution, and extension. Section III shows the experimental results on face classification, palm print classification, background modeling, and challenging image classification. Section IV concludes this paper and describes further works.

II. δ-Norm and Robust Regression Model

This section presents the form of δ-norm by generalizing the original l_0-norm. Afterward, we apply δ-norm to quantify the loss function residual with a kernel improvement, and propose the DRR. Then, a concrete kernel function is presented to obtain the closed-form solution to DRR. The proposed model is also extended to a multiscale version to further handle the mixed noise.

A. Definition of δ-Norm

The noise of the outlier is the main obstacle that regression models encountered to achieve satisfactory performance [51]. In practice, if an image is contaminated by sparse noise, the corrupted parts are then remarkably different from the original pixels. Consequently, the fitting residual will be significantly amplified by l_1-norm or l_2-norm. To avoid this serious impact caused by the corrupted data, we directly consider the count of the noisy elements. As an example, if we use the l_0-norm to characterize residual, the residual value will be the number of noisy elements in the vector, which weakens the influences caused by the outlier noise. However, the l_0-norm ignores the local correlations within a data example. To overcome this shortcoming, we extend the element form of l_0-norm to the neighborhood form by the following definition.

Definition 1: For a vector $\boldsymbol{\alpha} \in \mathbb{R}^d$, the ith neighborhood block is defined as

$$\boldsymbol{\delta}_i(\boldsymbol{\alpha}) = (\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+t})^\top \tag{1}$$

where $1 \le i \le d - t$ and $t < d$ is the neighborhood size parameter.

According to Definition 1, the ith neighborhood block not only contains the isolated single element but also includes its neighborhood information, that is, the neighborhood blocks take the local correlations of a vector into account. Based on the above definition, we further present the calculation form of the δ-norm[2] in Definition 2.

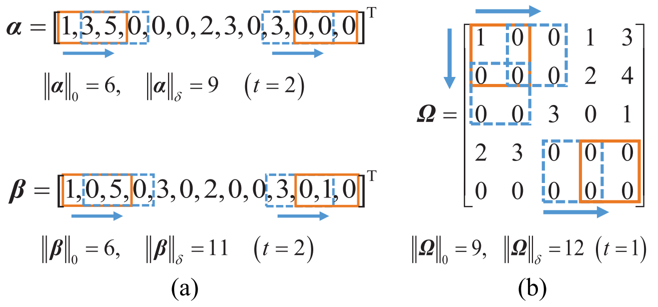

$\boldsymbol{\alpha} = [1,3,5,0,0,0,2,3,0,3,0,0,0]^\top$
$\|\boldsymbol{\alpha}\|_0 = 6$, $\|\boldsymbol{\alpha}\|_\delta = 9$ $(t=2)$
$\boldsymbol{\beta} = [1,0,5,0,3,0,2,0,0,3,0,1,0]^\top$
$\|\boldsymbol{\beta}\|_0 = 6$, $\|\boldsymbol{\beta}\|_\delta = 11$ $(t=2)$
(a)
$\boldsymbol{\Omega} = \begin{bmatrix} 1 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 3 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$\|\boldsymbol{\Omega}\|_0 = 9$, $\|\boldsymbol{\Omega}\|_\delta = 12$ $(t=1)$
(b)

Fig. 2. Examples for δ-norm calculations. (a) Vector form. (b) Matrix form. The neighborhood block slides from the first element to the last element, and the nonzero neighborhood blocks are counted. The neighborhood size $t = 0$ incurs $\|\boldsymbol{\alpha}\|_\delta = \|\boldsymbol{\alpha}\|_0$.

Definition 2: The δ-norm $\|\cdot\|_\delta$ for a vector $\boldsymbol{\alpha} \in \mathbb{R}^d$ is defined as

$$\|\boldsymbol{\alpha}\|_\delta = \sum_{i=1}^{d-t} \operatorname{sign}(\|\boldsymbol{\delta}_i(\boldsymbol{\alpha})\|_0) \tag{2}$$

where the l_0-norm $\|\cdot\|_0$ represents the number of nonzero elements in the vector.

As illustrated in Fig. 2, the δ-norm is based on the sliding of neighborhood blocks in the data examples. It is noteworthy that the l_0-norm is a special case of the δ-norm, that is, when the neighborhood size $t = 0$, we have

$$\|\boldsymbol{\alpha}\|_\delta = \sum_{i=1}^{d} \operatorname{sign}(\|\boldsymbol{\delta}_i(\boldsymbol{\alpha})\|_0) = \sum_{i=1}^{d} \operatorname{sign}(\|\alpha_i\|_0) = \|\boldsymbol{\alpha}\|_0. \tag{3}$$

From the perspective of residual measurements, if we adopt the δ-norm to characterize the distance between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, $\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_\delta$ equals the number of different neighborhood blocks between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Regardless of what the residual value is, $\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_\delta$ only counts the different neighborhood blocks based on the l_0-norm and, thus, suppresses the severe noises effectively. Meanwhile, our proposed δ-norm is a block-wise metric rather than the element-wise metrics, so it is very suitable for image examples considering the correlations between the adjacent feature elements. It means that elements in a neighborhood block are regarded as a whole and, thus, it can be structurally recovered in regression models. Compared to the other residual metrics which relax l_0-norm, our δ-norm is a direct extension to l_0-norm and can be solved precisely by our provided kernel method.

It is worth pointing out that the vector and matrix form of δ-norm have the same intrinsic properties. Therefore, we only consider the vector form throughout this paper for simplicity.

B. δ-Norm-Based Robust Regression

It is well known that the linear regression with Tikhonov regularization [23] is formatted as

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2 \tag{4}$$

which has been widely applied in machine learning for several tasks [47], [53], [55]. However, such an elementary model may not be robust to severe noise, because the pairwise distance

[2] Our δ-norm and some common metrics (e.g., l_0-norm, capped-l_1-norm, and low-rank function) are actually pseudonorms. They satisfy the non-negative and triangle conditions, but they are all nonlinear.

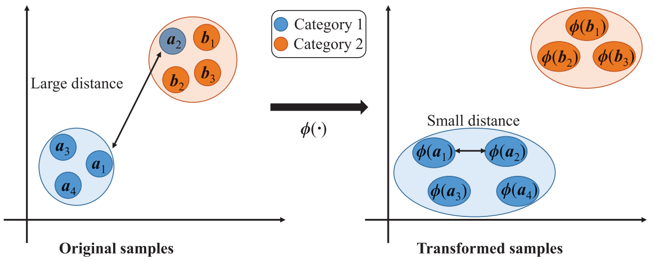


Fig. 3. Purpose of kernel mapping $\boldsymbol{\phi}$: data points are mapped to the kernel space in which the severe noise is suppressed effectively, and the regression is performed in the kernel space to obtain the robust representation result.

and reconstruction residual are both measured by the l_2-norm. Considering the importance of distance metrics in regression models, we apply the proposed δ-norm to quantify the regression residual for validating the performance of the proposed metric. Then, the primitive RR has a form of

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\delta + \lambda\|\boldsymbol{x}\|_2^2. \tag{5}$$

As the same statements in most regression models, the dictionary $\boldsymbol{A} = (\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n) \in \mathbb{R}^{d\times n}$ is composed of the training examples, and the vectors $\boldsymbol{b} \in \mathbb{R}^d$ and $\boldsymbol{x} \in \mathbb{R}^n$ are the test example and its corresponding representation coefficients, respectively. The reconstruction $\boldsymbol{A}\boldsymbol{x}$ recovers a clean pattern for the test example $\boldsymbol{b}$. Moreover, the regularization term $\|\boldsymbol{x}\|_2^2$ in the objective function is used to avoid overfitting for overcomplete dictionary [34].

Nevertheless, model in (5) has two adverse points for image analysis tasks. First, (5) is nonconvex and discontinuous. To the best of our knowledge, no algorithms are available to solve it effectively. Another point is that the regression result is unreasonable when the training examples in $\boldsymbol{A}$ are corrupted, because the recovered data $\boldsymbol{A}\boldsymbol{x}$ is still corrupted by the corrupted examples.

To overcome the aforementioned difficulties, we assume that there exists a kernel mapping $\boldsymbol{\phi}(\cdot)$ in Fig. 3, by which the serious noise in the example can be suppressed. To this end, we follow the common settings in the kernel learning algorithms [18], [20], and enforce the Euclidean distance in the assumed kernel space to be equivalent to the δ-norm distance in the original sample space. Therefore, mathematically, we construct the following set of kernel mapping for δ-norm:

$$\begin{aligned}\mathbb{D}(\delta) = \Big\{ \boldsymbol{\phi}_\delta^p : \mathbb{R}^d \mapsto \mathbb{R}^h \mid \lim_{p\to+\infty} \left\|\boldsymbol{\phi}_\delta^p(\boldsymbol{\alpha}) - \boldsymbol{\phi}_\delta^p(\boldsymbol{\beta})\right\|_2^2 \\ = \theta\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_\delta, \ \forall \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^d \Big\}\end{aligned} \tag{6}$$

where θ is a nonzero constant. Importantly, $\mathbb{D}(\delta)$ can be proved to be a nonempty set in the optimization. Based on the kernel mapping, the regression residual of (5) is equal to

$$\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\delta = \frac{1}{\theta}\lim_{p\to+\infty} \left\|\boldsymbol{\phi}_\delta^p(\boldsymbol{A}\boldsymbol{x}) - \boldsymbol{\phi}_\delta^p(\boldsymbol{b})\right\|_2^2. \tag{7}$$

Then, to improve the regression results in the kernel space, we replace $\boldsymbol{\phi}_\delta^p(\boldsymbol{A}\boldsymbol{x})$ with $\boldsymbol{\phi}_\delta^p(\boldsymbol{A})\boldsymbol{x}$,[3] because $\boldsymbol{\phi}_\delta^p(\boldsymbol{A})\boldsymbol{x}$ is able to represent any examples in the kernel space while $\boldsymbol{\phi}_\delta^p(\boldsymbol{A}\boldsymbol{x})$ can only cover a subset of examples in the kernel space [2]. Meanwhile, the noises of corrupted training examples in $\boldsymbol{A}$ are well suppressed by δ-norm after the operation of mapping $\boldsymbol{\phi}_\delta^p$ and, thus, the recovered result $\boldsymbol{\phi}_\delta^p(\boldsymbol{A})\boldsymbol{x}$ is more reliable than $\boldsymbol{\phi}_\delta^p(\boldsymbol{A}\boldsymbol{x})$. Hence, the kernel space regression $\boldsymbol{\phi}_\delta^p(\boldsymbol{A})\boldsymbol{x}$ and kernel mapping set $\mathbb{D}(\delta)$ are employed to improve (7). Our DRR is ultimately expressed as

$$\begin{aligned}&\min_{\boldsymbol{x}} \ f_p(\boldsymbol{x}) = \left\|\boldsymbol{\phi}_\delta^p(\boldsymbol{A})\boldsymbol{x} - \boldsymbol{\phi}_\delta^p(\boldsymbol{b})\right\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2 \\ &\ \text{s.t.} \ \ \boldsymbol{\phi}_\delta^p \in \mathbb{D}(\delta)\end{aligned} \tag{8}$$

where the dictionary $\boldsymbol{\phi}_\delta^p(\boldsymbol{A}) = (\boldsymbol{\phi}_\delta^p(\boldsymbol{A}_1), \boldsymbol{\phi}_\delta^p(\boldsymbol{A}_2), \ldots, \boldsymbol{\phi}_\delta^p(\boldsymbol{A}_n)) \in \mathbb{R}^{h\times n}$ and $\boldsymbol{x} \in \mathbb{R}^n$. According to the constraint of $\mathbb{D}(\delta)$ in (6), the infinite p value induced objective function $f(\boldsymbol{x}) = \lim_{p\to+\infty} f_p(\boldsymbol{x})$ leads to the strict equivalence of the Euclidean distance in the kernel space and the δ-norm distance in the original space. Although the pairwise δ-norm distance can be approximated infinitely when p is sufficiently large, it is noticed that (5) and (8) are not equivalent while the original regression is transformed to the kernel form. It means that δ-norm is employed in our DRR [i.e., (8)] to measure the pairwise distance and not merely the regression residual.

In Section II-C, we derive a concrete kernel mapping after some simplifications for (8). We also show that the mapping operations in the high-dimensional space can be avoided by kernel tricks and, thus, a closed-form solution is obtained.

C. Model Solution in the Kernel Space

An efficient solution is very important for an RR model, as the multiple iterations could consume lots of time, especially on the large-scale dataset [3]. For real-time computations, we derive a concrete kernel function to obtain a closed-form solution to DRR. When t is fixed to 0, the provided algorithm can also be seen as a more accurate solution to l_0-norm in the kernel space, compared with the other approximations to l_0-norm [15], [17].

It seems that (8) is much more difficult than (5) to solve, due to the additional unknown mapping. However, this unknown mapping helps us to eliminate the nonconvex and discontinuous calculations from δ-norm. Therefore, it is necessary to demonstrate that the kernel mapping $\boldsymbol{\phi}_\delta$ exists. We prove the existence by the following proposition.

Proposition 1: For all vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta} \in \mathbb{R}^d$, we have the following analytic function:

$$\widetilde{\kappa}_\delta^p(\boldsymbol{\alpha}, \ \boldsymbol{\beta}) = \sum_{i=1}^{d-t} \frac{1}{2^p}\left(\cos\left(\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha}), \ \hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right)+1\right)^p \tag{9}$$

[4]which is a kernel function and satisfies

$$\widetilde{\boldsymbol{\phi}}_\delta^p \in \mathbb{D}(\delta) \tag{10}$$

where t is the neighborhood size parameter of δ-norm and $\widetilde{\boldsymbol{\phi}}_\delta^p$ is the kernel mapping corresponding to $\widetilde{\kappa}_\delta^p$, that is, $\widetilde{\boldsymbol{\phi}}_\delta^p(\boldsymbol{\alpha})^\top \widetilde{\boldsymbol{\phi}}_\delta^p(\boldsymbol{\beta}) = \widetilde{\kappa}_\delta^p(\boldsymbol{\alpha}, \ \boldsymbol{\beta})$.

[3] Usually, $\boldsymbol{\phi}(\boldsymbol{A}\boldsymbol{x})$ is not equivalent to $\boldsymbol{\phi}(\boldsymbol{A})\boldsymbol{x}$ for the nonlinear mapping $\boldsymbol{\phi}$.

[4] Here, the vector augmentation operator $\hat{}$ indicates $\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha}) = (\boldsymbol{\delta}_i(\boldsymbol{\alpha})^\top, 1)^\top = (\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+t}, 1)^\top$.

The proof of the proposition is given in the Appendix. This proposition implies that the set $\mathbb{D}(\delta) \neq \emptyset$, that is, (8) is not ill-posed. It is worth pointing out that there might be more than one kernel satisfying (10), and here we just present a typical form to demonstrate its existence. Furthermore, the proof also suggests that the provided kernel function satisfies

$$\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_\delta = d - t - \lim_{p\to+\infty} \widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta}). \tag{11}$$

It means that we can present an analytic function

$$\Delta^p(\boldsymbol{\alpha},\ \boldsymbol{\beta}) = d - t - \widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta}) \tag{12}$$

which is an effective approximation for $\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_\delta$ when we set the parameter p to a large value. To visualize the approximation curves, Fig. 4 plots the 1-D case of (12) with different p values.

In the following discussion, we solve the proposed DRR model based on Proposition 1. Specifically, we replace the high-dimensional mapping in (8) with $\widetilde{\boldsymbol{\phi}}^p_\delta$ corresponding to (9), and the residual term in (8) is equal to

$$\begin{aligned}\left\|\widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{A})\boldsymbol{x} - \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{b})\right\|_2^2 &= \boldsymbol{x}^\top \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{A})^\top \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{A})\boldsymbol{x} \\ &\quad - 2\boldsymbol{x}^\top \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{A})^\top \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{b}) + c \\ &= \boldsymbol{x}^\top \boldsymbol{K}^p_{\boldsymbol{A}}\boldsymbol{x} - 2\boldsymbol{K}^p_{\boldsymbol{bA}}\boldsymbol{x} + c \end{aligned} \tag{13}$$

in which

$$\left(\boldsymbol{K}^p_{\boldsymbol{A}}\right)_{ij} = \widetilde{\kappa}^p_\delta\left(\boldsymbol{A}_i,\ \boldsymbol{A}_j\right),\ \ i, j = 1, 2, \ldots, n \tag{14}$$

and

$$\left(\boldsymbol{K}^{(p)}_{\boldsymbol{bA}}\right)_i = \widetilde{\kappa}^p_\delta(\boldsymbol{b},\ \boldsymbol{A}_i),\ \ i = 1, 2, \ldots, n \tag{15}$$

and $\widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta})$ is defined by (9). Consequently, the objective function in (8) is equal to

$$f_p(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{K}^p_{\boldsymbol{A}}\boldsymbol{x} - 2\boldsymbol{K}^p_{\boldsymbol{bA}}\boldsymbol{x} + \lambda\|\boldsymbol{x}\|_2^2 + c. \tag{16}$$

The kernel matrix $\boldsymbol{K}_A$ is positive semidefinite, so $f_p(\boldsymbol{x})$ is convex and the stationary point is the optimal point. Therefore, we let

$$\nabla f_p(\boldsymbol{x}) = 2\boldsymbol{K}^p_{\boldsymbol{A}}\boldsymbol{x} - 2\left(\boldsymbol{K}^p_{\boldsymbol{bA}}\right)^\top + 2\lambda\boldsymbol{I}_n\boldsymbol{x} = 0 \tag{17}$$

and obtain that the closed-form solution to (8) is

$$\boldsymbol{x}^*_p = \left(\boldsymbol{K}^p_{\boldsymbol{A}} + \lambda\boldsymbol{I}_n\right)^{-1}\left(\boldsymbol{K}^p_{\boldsymbol{bA}}\right)^\top. \tag{18}$$

Meanwhile, when p is close to $+\infty$, we have

$$\boldsymbol{x}^* = \lim_{p\to+\infty} \boldsymbol{x}^*_p = \lim_{p\to+\infty} \arg\min f_p(\boldsymbol{x}) = \arg\min f(\boldsymbol{x}) \tag{19}$$

which gives rise to the strict δ-norm distance characterization of DRR.

Empirically, we achieve a sufficient approximation in (19) when the exponent $p \geq 20$. In this case, the numerical calculations are not ill-posed problems. Furthermore, this approximation is actually a blessing for the residual metric, because the varied exponent improves the smoothness of δ-norm which avoids the discrete loss values. The detailed steps of solving DRR are summarized in Algorithm 1.

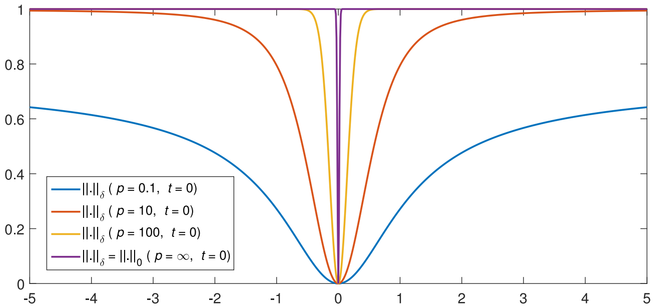


Fig. 4. Approximation curves of δ-norm in 1-D case with different parameters. The neighborhood size t in (12) is fixed as 0 for visualization, and the horizontal axis records the difference between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The infinite p value renders the strict equivalence to δ-norm.

```
Algorithm 1 Solving Representation Coefficients for DRR
Input: training examples $A_1, ..., A_n$; test example $b$.
Parameters: $\lambda > 0$, $p \geq 20$ and $t \in \mathbb{N}$.
Initialization: compute $M = (K^p_A + \lambda I_n)^{-1}$ by Eq. (14).
Procedure:
  1) compute $v = (K^p_{bA})^\top$ by Eq. (15);
  2) compute $x^* = Mv$.
End
Output: representation coefficients $x^*$.
```

The main time consumption in Algorithm 1 is the calculation for inverting a matrix. Interestingly, this matrix inversion can be computed in training phase, and accelerated by various numerical methods (e.g., Nystrom approximation) [57]. Hence, the algorithm only needs linear projections and kernel vector calculations with a total complexity of $O(n^2 + nd)$, which is more efficient than other iteration-based regression models. Moreover, when GPU is employed, the complexity can be reduced to at least efficient $O(n + d)$ [8].

D. DRR-Based Classifier and Reconstruction

The classification can be easily performed based on the obtained representation. For a test example $\boldsymbol{b}$ and the corresponding representation $\boldsymbol{x}^*$ obtained from Algorithm 1, its label is

$$\text{Label}(\boldsymbol{b}) = \text{Label}\left(\boldsymbol{A}_j\right) \tag{20}$$

where $j = \arg\min_i \|\boldsymbol{x}^* - \boldsymbol{e}_i\|$ and $\boldsymbol{e}_i = [0, 0, \ldots, 1, \ldots, 0, 0]^\top$ is the ith orthonormal basis in $\mathbb{R}^n$. The label decision is actually accomplished by KNN, which calculates the distances between the representation vector and the ith orthonormal basis. It is worth pointing out that δ-norm can be used in other conventional discriminant classifiers, such as kernel SVM [38] and kernel logistic regression [60]. After that, we can obtain the robust version of these classifiers, in which the pairwise distances between examples are measured by δ-norm.

However, similar to other kernel-based models, DRR is unable to directly recover the reconstruction $\boldsymbol{y}^*$ for the test example, because the kernel mappings for the kernel functions are not always invertible [25]. Hence, we reconstruct the test example by synthesizing the obtained representation and ridge regression. More specifically, the representation coefficients $\boldsymbol{x}^*$ are utilized to supervise the ridge regression results, which make the final representation $\hat{\boldsymbol{x}}$ to be close to it. That is

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda' \left\|\boldsymbol{x} - \boldsymbol{x}^*\right\|_2^2 \\ &= \left(\boldsymbol{A}^\top \boldsymbol{A} + \lambda' \boldsymbol{I}_n\right)^{-1} \left(\boldsymbol{A}^\top \boldsymbol{b} + \lambda' \boldsymbol{x}^*\right) \end{aligned} \tag{21}$$

and the final reconstruction is

$$\boldsymbol{y}^* = \boldsymbol{A}\hat{\boldsymbol{x}}. \tag{22}$$

E. Multiscale Form of DRR

As we described in Definition 2, the proposed δ-norm is intrinsically a series of metrics with varied scales. Consider that different neighborhood block sizes are effective in suppressing different severe noise, for example, element impulsive noise and block occlusion noise. To further handle the mixed noise cases, this section generalizes the proposed DRR to a multiscale form called MDRR, in which the regression loss is characterized by multiscale δ-norm with different parameters.

In our following descriptions, $\delta_1, \delta_2, \ldots, \delta_m$ indicate the norm with m different scales, and $t_1, t_2, \ldots, t_m$ are the corresponding neighborhood size values. The MDRR is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \ & F_p(\boldsymbol{x}) = \sum_{r=1}^{m} \left\| \boldsymbol{\phi}_{\delta_r}^p(\boldsymbol{A})\boldsymbol{x} - \boldsymbol{\phi}_{\delta_r}^p(\boldsymbol{b}) \right\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2 \\ \text{s.t.} \ & \boldsymbol{\phi}_{\delta_r}^p \in \mathbb{D}(\delta_r), \ \ r = 1, 2, \ldots, m \end{aligned} \tag{23}$$

in which each δ_r corresponds to a kernel mapping $\boldsymbol{\phi}_{\delta_r}^p$, and $\mathbb{D}(\delta_r)$ is the mapping set decided by δ_r-norm. The extended problem invokes m kernel mappings and, thus, we solve (23) by using Proposition 1 m times. Specifically, the kernel function corresponding to $\boldsymbol{\phi}_{\delta_r}$ is expressed as

$$\widetilde{\kappa}_{\delta_r}^p(\boldsymbol{\alpha}, \ \boldsymbol{\beta}) = \sum_{i=1}^{d-t_r} \frac{1}{2^p} \left(\cos\left(\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha}), \ \hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right) + 1 \right)^p \tag{24}$$

and we have

$$\begin{aligned} F_p(\boldsymbol{x}) &= \sum_{r=1}^{m} \left\| \widetilde{\boldsymbol{\phi}}_{\delta_r}(\boldsymbol{A})\boldsymbol{x} - \widetilde{\boldsymbol{\phi}}_{\delta_r}(\boldsymbol{b}) \right\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2 \\ &= \boldsymbol{x}^\top \left(\sum_{r=1}^{m} \widetilde{\boldsymbol{\phi}}_{\delta_r}^p(\boldsymbol{A})^\top \widetilde{\boldsymbol{\phi}}_{\delta_r}^p(\boldsymbol{A}) \right) \boldsymbol{x} \\ &\quad - 2\boldsymbol{x}^\top \sum_{r=1}^{m} \widetilde{\boldsymbol{\phi}}_{\delta_r}^p(\boldsymbol{A})^\top \widetilde{\boldsymbol{\phi}}_{\delta_r}^p(\boldsymbol{b}) + \sum_{r=1}^{m} \widetilde{\kappa}_{\delta_r}^p(\boldsymbol{b}, \ \boldsymbol{b}) + \lambda \|\boldsymbol{x}\|_2^2 \\ &= \boldsymbol{x}^\top \left(\sum_{r=1}^{m} \boldsymbol{K}_{\boldsymbol{A}}^{p, \ r} \right) \boldsymbol{x} - 2\boldsymbol{x}^\top \sum_{r=1}^{m} \boldsymbol{K}_{\boldsymbol{bA}}^{p,r} + \lambda \|\boldsymbol{x}\|_2^2 + c'. \end{aligned} \tag{25}$$

Therefore, the gradient condition gives

$$\Delta F_p(\boldsymbol{x}) = 2 \left(\sum_{r=1}^{m} \boldsymbol{K}_{\boldsymbol{A}}^{p, \ r} \right) \boldsymbol{x} - 2 \left(\sum_{r=1}^{m} \boldsymbol{K}_{\boldsymbol{bA}}^{p,r} \right)^\top + 2\lambda \boldsymbol{x} = 0 \tag{26}$$

and thus

$$\boldsymbol{x}_p^* = \left(\sum_{r=1}^{m} \boldsymbol{K}_{\boldsymbol{A}}^{p,r} + \lambda \boldsymbol{I}_n \right)^{-1} \left(\sum_{j=1}^{m} \boldsymbol{K}_{\boldsymbol{bA}}^{p,r} \right)^\top \tag{27}$$

in which

$$\left(\boldsymbol{K}_{\boldsymbol{A}}^{p,r}\right)_{ij} = \widetilde{\kappa}_{\delta_r}^p\left(\boldsymbol{A}_i, \ \boldsymbol{A}_j\right), \ \ i, j = 1, 2, \ldots, n \tag{28}$$

and

$$\left(\boldsymbol{K}_{\boldsymbol{bA}}^{p,r}\right)_i = \widetilde{\kappa}_{\delta_r}^p(\boldsymbol{b}, \ \boldsymbol{A}_i), \ \ i = 1, 2, \ldots, n. \tag{29}$$

From the above calculations, it can be easily grasped that MDRR can be divided into two phases, that is, the training phase and the test phase. The training phase learns a projection matrix via the provided dictionary (or original training examples), and the test phase calculates the representation coefficients based on the learned matrix. Specifically, we summarize the training and test phases in Algorithms 2 and 3. The two phases share the similar properties with Algorithm 1, and differ in the additional loops for kernel matrices calculation. After obtaining the representation coefficients $\boldsymbol{x}^* = \boldsymbol{x}_p^*$, the classification and reconstruction tasks of MDRR are the same as the forms of DRR in Section II-D.

Algorithm 2 Computing the Projection Matrix for MDRR

```
Input: training examples $A_1, ..., A_n$.
Parameters: $\lambda > 0$, $p \geq 20$ and scales $t_1, t_2, ..., t_m \in \mathbb{N}$ .
Initialization: index $r = 1$, projection matrix $\widetilde{M} = 0$.
While $r \leq m$
    1) compute $T = K_{A}^{p,\ r}$ by Eq. (28);
    2) compute $\widetilde{M} = \widetilde{M} + T$;
    3) $r = r + 1$.
End
Compute $M = (\widetilde{M} + \lambda I_n)^{-1}$.
Output: projection matrix $M$.
```

Algorithm 3 Test Phase for MDRR

```
Input: test example $b$; learned projection matrix $M$.
Parameters: $p \geq 20$ and scales $t_1, t_2, ..., t_m \in \mathbb{N}$ .
Initialization: index $r = 1$, vector $v = 0$.
While $r \leq m$
    1) compute $w = K_{bA}^{p,\ r}$ by Eq. (29);
    2) compute $v = v + w$;
    3) $r = r + 1$.
End
Compute $x^* = Mv$.
Output: representation coefficients $x^*$.
```

III. Experiments

This section evaluates the proposed DRR from several aspects. Section III-A tests the pairwise distance of examples in the kernel space, and explores the representation distributions of DRR, to explain why DRR is robust to the corrupted example. Section III-B compares DRR with some state-of-the-art representation-based classifiers, including SRC [46], CRC [55], RSC [52], Capped-l_1-DL [15], SSRR [45], ProCRC [5], NMR [51], and ACPRC [32] on corrupted biometrics recognition, using Extended-YaleB [22], AR [30], Robust-NUST,[5] and PolyU palm print [54] datasets.

[5] The Robust-NUST dataset is available at https://github.com/functioncs/Robust-NUST.

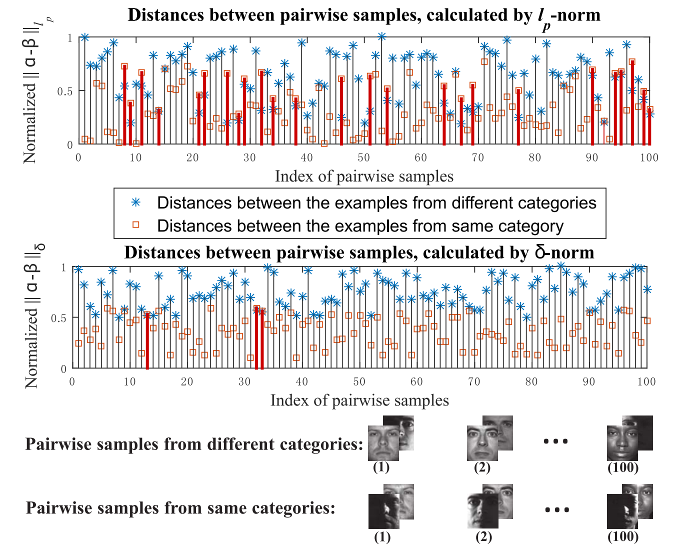
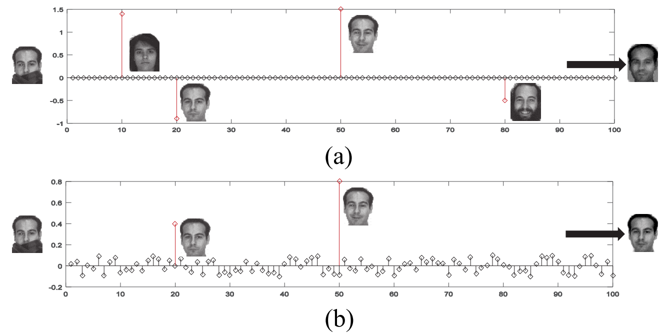


Fig. 5. Distance distributions for pairwise examples, which are calculated by l_p-norm (l_1-norm as an example) and δ-norm (with $t = 3$): each pair contains two examples from either different categories or the same category. The selected examples are corrupted by severe illumination noise. The red lines denote incorrect metric results, namely, the distance of the examples in the same category is greater than the distance of examples belonging to different categories.


(a)

(b)

Fig. 6. Comparison between the l_p-norm-based regression (SRC as an example), and the proposed DRR. The red lines denote the significant impulses in the representation coefficients. (a) SRC representation coefficients of the corrupted sample with scarf disguise. (b) DRR representation coefficients of the corrupted sample with scarf disguise.

In Section III-C, we compare DRR with representative projection-based methods, including PCA [16], robust-PCA (RPCA) [6], Euler-PCA [28], PCA-graph [39], and RR [12] on the background modeling dataset [24]. It should be mentioned that the representation-based methods are not applicable to background modeling tasks and the projection-based methods are not applicable to classifications of corrupted examples. Hence, we only compare these two methods on their studied tasks, respectively. In Section III-D, we evaluate DRR on the object classification dataset Caltech-256 [9] and large-scale classification dataset ImageNet ILSVRC 2012 [37]. In Section III-E, we evaluate the running time of DRR and MDRR. Finally, in Section III-F, we test the parametric sensitivity of DRR and MDRR.

The proposed DRR has three parameters, that is, t, p, and λ. In the following experiments, we set $t \in \{3,\ 6,\ 9\}$, $p = 30$, and $\lambda \in \{0.5,\ 1,\ 1.5\}$. In addition, the multiscale parameters t_i $(i = 1, \ldots, m)$ in MDRR are fixed to $t_1 = 1$, $t_2 = 2$, $t_3 = 4$, and $t_4 = 8$. For other classifiers, their parameters are tuned to achieve the best results in each experiment. In our experiments, images are resized to 32×32, so we simply set the neighborhood size t within a proper range. Moreover, it is observed from Fig. 16 that the performances of DRR is stable when $p > 20$ and $\lambda \in (0.1,\ 2)$, so we directly choose parameters in such ranges for our experiments.

A. *Exploration for the Kernel Space and Representation Coefficients*

As mentioned before, the proposed kernel space measures the pairwise distance between examples by δ-norm, so we select a series of pairwise examples from Extended-YaleB dataset with severe illumination changes. It is worth pointing out that the δ-norm distance between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is calculated by $\Delta^p(\boldsymbol{\alpha},\ \boldsymbol{\beta})$ in (12), because the proposed δ-norm is finally converted to its approximation form with a tuned p value. (Here, p is fixed to 30.) In Fig. 5, the intraclass variations from corrupted examples are significantly suppressed by the δ-norm metric and, thus, the incorrect measure results are reduced. It means that the kernel $\boldsymbol{\phi}(\cdot)$ corresponds to a limited feature space, by which the noises can be suppressed effectively.

Besides, the classification and reconstruction of DRR finally depend on the calculated representation coefficients, so we manage to explore the distributions of coefficients. Here, we reconstruct a corrupted face image with scarf disguise, and visualize the corresponding representation coefficients. As shown in Fig. 6, the representation coefficients of SRC have several high impulses on other categories, despite zero impulses. In contrast, DRR representation coefficients of the test example have high impulses on its own category examples, while have very low impulses on the examples of other categories. This is because the influence of corruption is reduced by δ-norm, allowing for the examples of the same category to stay together. Accordingly, the corrupted data will only be fitted on the examples of the same category and, thus, DRR also generates a more reasonable reconstructed image.

B. *Biometrics Image Recognitions With Corruptions*

In this experiment, we consider four types of corruptions: 1) artificial occlusion; 2) real disguise; 3) random pixel noise; and 4) illumination changes corruption. Four popular biometrics datasets are utilized.

Extended-YaleB dataset contains *face images* of 38 persons under 9 poses and 64 illumination conditions. We select the noncorrupted examples in subset 1 for training, and the examples with extreme illumination changes in subsets 4 and 5 are used for testing. Examples from subset 3 are artificially corrupted by random sparse noise for additional testing, which is shown in Fig. 7.

AR dataset contains over 4000 color *face images* of 126 people, including different facial expressions, lighting conditions, glasses occlusion, and scarf occlusion. We use the 960 nonoccluded images for training, and use the 600 images with sunglasses and scarf for testing. Several nonoccluded images

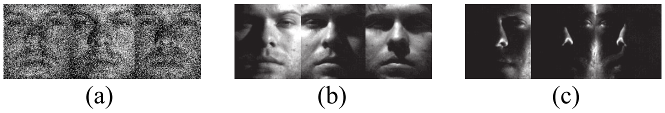


Fig. 7. Example images from the Extended-YaleB dataset. (a) Subset 3 with random noise. (b) Subset 4 with illumination changes. (c) Subset 5 with illumination changes.

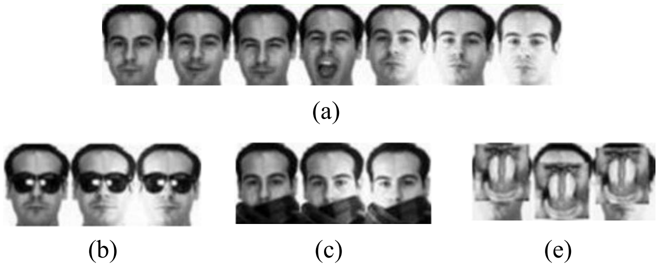


Fig. 8. Example images from the AR dataset. Examples with (a) expression changes, (b) sunglasses occlusions, and (c) scarf occlusions. (d) Examples corrupted by "baboon" block.

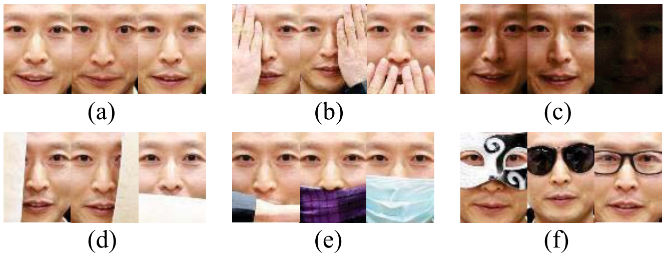


Fig. 9. Example images from the Robust-NUST dataset. (a) Examples without noise. Examples with (b) hand occlusions, (c) illumination changes, (d) book occlusions, (e) mouth disguise, and (f) eyes disguise.

are artificially occluded by Baboon block for additional testing. Fig. 8 shows the examples of one individual. Notice that in the test images with "baboon occlusion," only a local image patch is corrupted when compared with the clean training images and, thus, they do not have any global changes, such as background variations, expression variations, or illumination changes which often occur in the test images with "scarf occlusion." As a result, our algorithm with baboon occlusion produces higher accuracy than the scarf occlusion cases.

Robust-NUST dataset contains various types of corrupted *face images* of 200 subjects, such as scarf occlusion, glasses occlusion, hand occlusion, book occlusion, and illumination changes. Fig. 9 shows several examples for one person. We use the eight clean images from each subject for training, and all of the corrupted examples are employed for testing.

PolyU dataset contains 8000 examples of 400 *palm images* from 200 volunteers, including 136 males and 64 females. The age ranges from 10 to 55 years. As shown in Fig. 10, the PolyU dataset is divided into two sessions, and we use the images in the second session for testing, which are corrupted by artificial occlusions with the noise level ranging from 10% to 90%.

The accuracy results in Table II reveal that capped-l_1-DL, NMR, and DRR are superior to other regression models, because they have more reasonable residual metrics, which are able to suppress the severe noise. More specifically, with a more robust δ-norm, our DRR and MDRR render the best performances among all of the comparators. Note that the parameter t in DRR is carefully tuned to achieve the best performance, while MDRR simply sets the default neighborhood size to 4. The parameter-free method MDRR is able to achieve no worse results than the DRR with careful parameter tuning. Therefore, the effectiveness and reliability of MDRR can be verified.

Fig. 10. Example images from PolyU palmprint dataset. The session 2 images are occluded by tiger patches.

To further evaluate the performance of the proposed model under corrupted training examples, we randomly select training examples for all of the compared methods, and the remaining examples are used for testing. By following the settings in [41], both clean and corrupted data examples are used for training. We select 2, 4, 6, 8, and 10 training examples for experiments, and the corresponding experimental results are shown in Fig. 11. It is clear that our proposed DRR and MDRR achieve satisfactory results when the training examples are more than 6. We can also find that DRR and MDRR are superior to other regression methods by increasing the number training examples.

C. Background Modeling

We evaluate the reconstruction ability of DRR on the background modeling task. We use the popular dataset from Li *et al.* [24], which consists of nine videos, including illumination changes, indoor/outdoor environments, as well as dynamic background changes. The following similarity measure is used for quantitative results, namely:

$$\text{Similarity} = \frac{tp}{tp + fp + fn} \tag{30}$$

where tp, fp, and fn are the numbers of correctly labeled foreground, falsely labeled background, and falsely labeled foreground pixels, respectively [35].

In this experiment, we take the foreground as the structured noise, and the background as the example to be reconstructed. It should be mentioned that the huge training examples always lead to overfitting in regression models, which is very different from the dimension reduction methods. Therefore, we selected 20 frames in each video for training, and the frames with groundtruth labeling for testing. Since the traditional regression models cannot avoid overfitting, we select the unsupervised dimension reduction methods for comparisons. The extraction accuracy is listed in Table III, and two example results are shown in Fig. 12. We can see that the Euler-PCA and DRR yield the best performance among all compared methods, because they have more robust residual metrics than other methods.

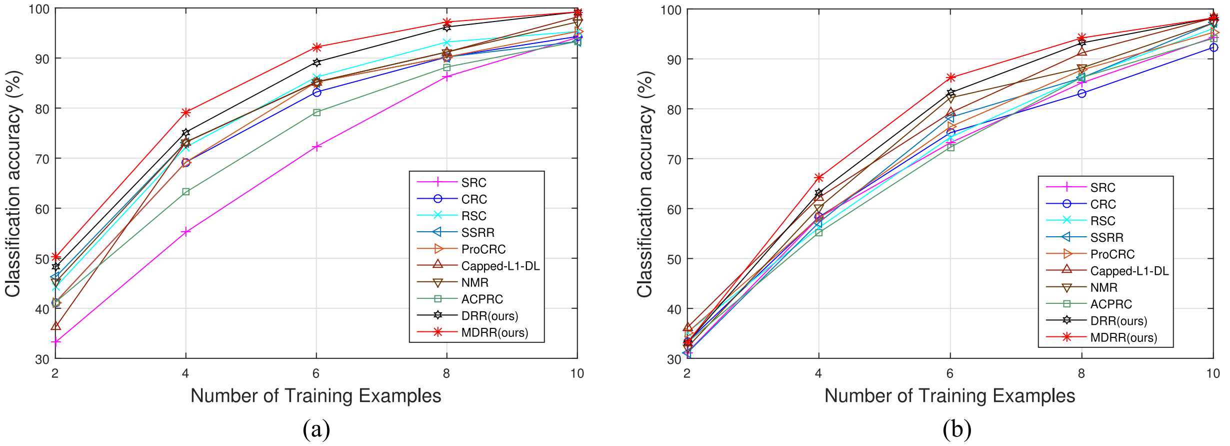


Fig. 11. Classification accuracy (%) under different numbers of training examples on: (a) AR and (b) Extended-YaleB datasets. The randomly selected training examples include clean and corrupted data.

TABLE II
CLASSIFICATION ACCURACY (%) ON EXTENDED-YALEB, AR, ROBUST-NUST, AND POLYU DATASETS, WITH VARIOUS CORRUPTIONS AND NOISES

Datasets	**Extended-YaleB**			**AR**			**Robust-NUST**					**PolyU**			
Corruption types	Random	Sub. 4	Sub. 5	Sun.	Scarf	Baboon	Hand	Book	Illum.	Eyes	Mouth	10%	30%	60%	90%
SRC [46]	93.6	78.4	28.8	94.4	57.6	70.0	64.7	46.7	36.3	34.3	11.7	99.1	85.2	75.7	28.8
CRC [55]	97.1	88.0	35.0	93.5	63.6	68.6	54.4	43.7	82.0	37.3	29.0	97.1	82.0	67.3	35.0
RSC [52]	91.3	80.3	36.7	94.8	66.8	91.8	75.3	79.3	21.3	39.3	17.0	98.3	80.3	66.3	36.7
SSRR [45]	96.5	88.1	45.3	95.0	63.5	88.9	83.9	73.0	27.7	46.0	17.3	98.1	82.2	81.1	45.3
ProCRC [5]	97.3	89.2	41.3	94.5	69.3	78.2	65.5	60.7	83.4	45.6	40.6	98.3	83.2	69.6	41.3
Capped-l_1-DL [15]	98.2	89.3	48.4	95.4	66.7	92.0	70.6	52.7	83.7	43.7	29.7	98.2	88.3	83.7	55.3
NMR [51]	98.3	90.2	47.9	96.9	73.5	95.1	88.7	87.7	84.0	52.0	51.0	**99.3**	90.2	82.0	47.9
ACPRC [32]	98.3	85.3	39.7	95.9	69.1	72.1	69.3	79.3	68.4	51.1	59.1	99.1	87.3	81.3	45.4
DRR (ours)	**99.3**	**93.0**	**60.4**	**98.5**	**86.5**	**99.3**	**94.0**	**95.0**	**92.3**	**75.3**	**75.3**	**99.3**	**93.0**	**85.3**	**60.4**
MDRR(ours)	**99.3**	**95.3**	**65.1**	**98.6**	**86.9**	**99.6**	**95.2**	**96.3**	**93.5**	**78.3**	**76.5**	**99.3**	**93.1**	**85.3**	**60.4**

TABLE III
BACKGROUND MODELING ACCURACY (%) ON THE LI DATASET WITH SEVERAL REPRESENTATIVE SCENES

Scenes	Airport	Bar	Lobby	Curtain	Escalator	Fountain	Mall	Campus	Water	Average
PCA [16]	0.540	0.503	0.600	0.686	0.442	0.508	0.545	0.286	0.767	0.5419
RPCA [6]	0.540	0.504	0.604	0.671	0.442	0.548	0.540	0.286	0.767	0.5407
PCA-Graph [39]	0.564	0.499	0.607	0.727	0.428	0.563	0.581	0.294	0.693	0.5447
Euler-PCA [28]	0.574	0.533	0.615	0.755	0.479	**0.568**	0.564	0.292	0.774	0.5738
RR [12]	0.581	0.487	0.599	0.733	0.443	**0.571**	0.621	0.321	0.723	0.561
DRR (ours)	**0.584**	**0.562**	**0.625**	**0.767**	**0.482**	0.568	**0.615**	**0.325**	**0.776**	**0.5893**
MDRR(ours)	**0.589**	**0.564**	**0.651**	**0.769**	**0.487**	**0.573**	**0.617**	**0.420**	**0.779**	**0.591**

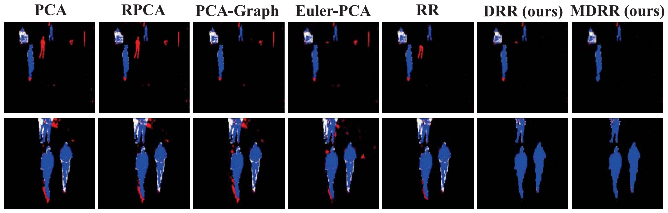


Fig. 12. Example results of background modeling. Black indicates the correctly predicted background; blue indicates the correctly predicted foreground; red indicates misclassified background; and white indicates misclassified foreground.

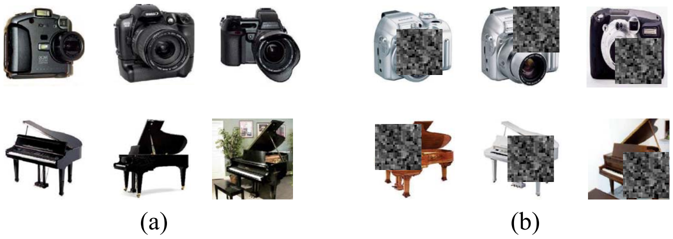


Fig. 13. Example images from the Caltech-256 dataset. The above row is the category of "camera," and the below row is the category of "piano," including (a) clean training examples and (b) corrupted test examples.

D. Object Classification

Caltech-256 dataset is composed of 256 object categories with 80–150 images per category. Fig. 13 shows the example images from two categories, in which the test examples are occluded by blocks of random noises. By following the common experimental settings, we randomly select 60 images from each category for training, and use the remaining images for testing. The general image classification is

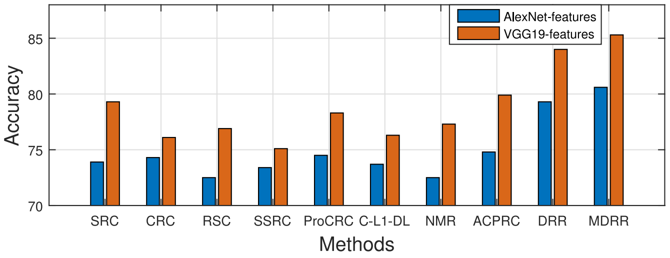


Fig. 14. Classification accuracy (%) on Caltech-256, using AlexNet-features and VGG19-features.

TABLE IV
Classification Accuracy (%) on ImageNet-2012

Features	BOW-SIFT		AlexNet		VGG19	
	TOP1	TOP5	TOP1	TOP5	TOP1	TOP5
SRC [46]	5.9	24.6	49.1	69.7	51.9	71.9
CRC [55]	6.3	24.3	50.3	68.5	53.3	73.3
RSC [52]	6.5	23.9	50.8	72.1	51.8	71.5
SSRR [45]	6.9	27.1	51.3	73.1	52.1	74.5
ProCRC [5]	7.1	25.6	51.3	71.0	55.5	76.7
Capped-l_1-DL [15]	6.9	26.3	52.7	73.1	55.1	76.9
NMR [51]	7.6	28.3	53.1	72.9	56.1	75.3
ACPRC [32]	7.5	26.3	50.8	76.7	56.1	76.8
DRR (ours)	**8.3**	**29.8**	**57.1**	**78.9**	**58.4**	**78.9**
MDRR(ours)	**8.5**	**30.2**	**57.9**	**79.2**	**58.9**	**81.3**

quite challenging, so we employ CNN image features [21], [40] rather than the grayscale features for all of the compared methods. Fig. 14 shows that the proposed method obtains the competitive performance with the CNN feature map. It is clear that DRR and MDRR perform more robustly than other regression methods.

ImageNet Large-Scale Visual Recognition Challenge 2012 dataset [37] consists of 1.2M+ training images from 1000 categories (about 1300 images per category) and 50K validation images. Similar to the experiments on Caltech-256, we also corrupted the validation images by using noise blocks before feature extractions. We compare DRR with other classifiers by using three different features: 1) BOW-SIFT features extracted from vlfeat [42]; 2) AlexNet [21]; and 3) VGG19 [40] features extracted by Caffe [14]. For each category, a dictionary with 50 atoms is learned from the training images. In the proposed DRR model, a matrix inversion operation is invoked to calculate the projection matrix. The inversion operation consumption depends on the dimension of training data. Therefore, it might be unrealistic to solve a matrix inversion with millions of dimensions. Accordingly, we here utilize the DL techniques, which learns a dictionary with a small number of atoms [59], and then the matrix of training examples is replaced with this learned dictionary matrix. The TOP1 and TOP5 classification accuracies are listed in Table IV. With AlexNet and VGG19 features, DRR has the highest accuracy both on TOP1 and TOP5 tasks.

E. Running Time Comparison

This section compares the running time of DRR with the state-of-the-art representation-based methods. All algorithms are implemented on a Core Duo 2.93-GHz desktop with 8-GB RAM.

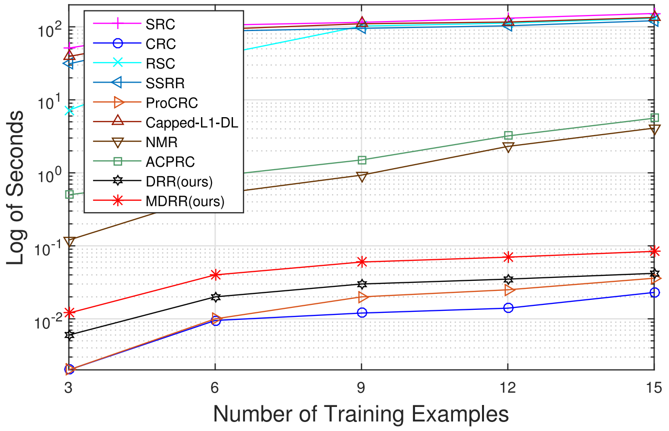


Fig. 15. Average running time (log of seconds) of classifying one test example on Extended-YaleB dataset.

TABLE V
Average Running Time (Seconds) of Classifying One Test Example on Caltech-256 and ImageNet-2012 Datasets

Datasets	Caltech-256	ImageNet-2012
SRC [46]	0.139	3.572
CRC [55]	0.0006	0.005
RSC [52]	0.097	3.499
SSRR [45]	0.151	4.573
ProCRC [5]	0.002	0.012
Capped-l_1-DL [15]	0.058	1.361
NMR [51]	0.012	0.274
ACPRC [32]	0.171	2.973
DRR (ours)	0.002	0.025
MDRR(ours)	0.003	0.032

Since the running time of regression models mainly depends on the dictionary size, we conduct the experiment in face recognition (Baboon, 50% occlusion level) on the Extended-YaleB dataset with varied training examples (dictionary atoms). The average running time (log of seconds) of recognizing one test example for each method is illustrated in Fig. 15. To further evaluate the efficiency of the proposed method on real image datasets, Table V presents the running time of compared methods on all experiments conducted in Section III-D. We can see that CRC, ProCRC, and DRR are the most efficient methods among all comparators as they all have closed-form solutions. Compared to the fastest model CRC, our DRR needs a little bit of additional time for building the kernel matrix.

F. Parametric Sensitivity

First, we evaluate the parametric sensitivity on the classification task. By changing the parameters in the Extend-YaleB experiment, Fig. 16 reveals that DRR is very robust to the variation of the regularization parameter λ, so this parameter can be easily tuned for practical use. Meanwhile, we can see that the performance of DRR is somewhat influenced by the neighborhood size t. This is because the $t > 0$ cases capture the local correlations in examples. The optimal t value

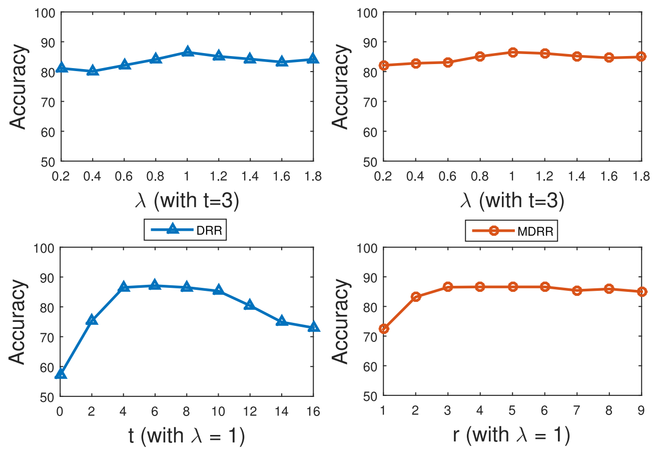


Fig. 16. Parametric sensitivity of DRR and MDRR. The first column shows the accuracy of DRR with varied regularization parameter λ and neighborhood size t. The second column evaluates the influence of the regularization parameter λ and the number of scales r.

is around 6, and the favorable results can be obtained when $t \in [4,\ 12]$. If the image examples suffer from severe illumination changes, the noises reveal the local correlations, so $t = 0$ cannot work well in such cases. It can also be found that MDRR has stable performance with the change of λ and r.

IV. Conclusion

This paper proposes a novel δ-norm to generalize the l_0-norm for residual characterization in regression. The DRR model is implemented for classification and reconstruction tasks. The theoretical analysis and experiments indicate that: 1) DRR is more robust than the l_p-norms-based regressions; 2) DRR is more efficient than the traditional RR models, because it has a closed-form solution; and 3) coupled with CNN features, DRR demonstrates the competitive performance in the challenging image classification tasks. In the future, we plan to adapt the proposed δ-norm to more reconstruction and regression models.

Appendix
Proof of Proposition 1

We first prove that (9) is a kernel function after some decomposition, and then we demonstrate the property of the kernel mapping. Specifically

$$\begin{aligned}\widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta}) &= \sum_{i=1}^{d-t}\frac{1}{2^p}\left(\cos\left(\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha}),\ \hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right)+1\right)^p\\ &= \sum_{i=1}^{d-t}\sum_{j=0}^{p}\frac{C_p^j}{2^p}\left(\frac{\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})^\top\hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})}{\left\|\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})\right\|\ \left\|\hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right\|}\right)^j\\ &= \sum_{i,\ j}\left(\sqrt{\frac{C_p^j}{2^p}}\,\frac{\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})^{(j)\top}}{\left\|\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})\right\|_2^j}\right)\left(\sqrt{\frac{C_p^j}{2^p}}\,\frac{\hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})^{(j)}}{\left\|\hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right\|_2^j}\right)\\ &= \operatorname{vec}\left(\bar{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\alpha})\right)^\top\operatorname{vec}\left(\bar{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\beta})\right)\\ &= \widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\alpha})^\top\widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\beta}) \qquad (31)\end{aligned}$$

where

$$\bar{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\alpha})_{ij:} = \sqrt{\frac{C_p^j}{2^p}}\,\frac{\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})^{(j)}}{\left\|\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})\right\|_2^j} \in \mathbb{R}^{t^j} \qquad (32)$$

and $\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})^{(j)}$ denotes the j-order Cartesian-product [3] of $\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha})$. Hence, (9) satisfies the definition of the kernel function. Furthermore, we have that $\forall\boldsymbol{\alpha},\ \boldsymbol{\beta} \in \mathbb{R}^d$

$$\lim_{p\to+\infty}\left(\frac{\cos\left(\hat{\boldsymbol{\alpha}},\ \hat{\boldsymbol{\beta}}\right)+1}{2}\right)^p = \begin{cases}1, & \text{if } \boldsymbol{\alpha} = \boldsymbol{\beta}\\ 0, & \text{if } \boldsymbol{\alpha} \neq \boldsymbol{\beta}\end{cases} \qquad (33)$$

because for any $\hat{\alpha}_i \neq \hat{\beta}_i$ $(i = 1, \ldots, d)$, the Cauchy-inequality [31] satisfies

$$\left(\sum_{i=1}^{d+1}\hat{\alpha}_i\hat{\beta}_i\right)^2 < \sum_{i=1}^{d+1}(\hat{\alpha}_i)^2\sum_{i=1}^{d+1}\left(\hat{\beta}_i\right)^2 \qquad (34)$$

by leveraging $\hat{\alpha}_{d+1} = \hat{\beta}_{d+1} = 1$. It means that

$$\operatorname{sign}\left(\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_0\right) = 1 - \lim_{p\to+\infty}\left(\frac{\cos\left(\hat{\boldsymbol{\alpha}},\ \hat{\boldsymbol{\beta}}\right)+1}{2}\right)^p. \qquad (35)$$

Then, (33) gives

$$\begin{aligned}\|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_\delta &= \sum_{i=1}^{d-t}\operatorname{sign}\left(\|\boldsymbol{\delta}_i(\boldsymbol{\alpha}-\boldsymbol{\beta})\|_0\right)\\ &= \sum_{i=1}^{d-t}\operatorname{sign}\left(\|\boldsymbol{\delta}_i(\boldsymbol{\alpha})-\boldsymbol{\delta}_i(\boldsymbol{\beta})\|_0\right)\\ &= \sum_{i=1}^{d-t}\left(1-\lim_{p\to\infty}\left(\frac{\cos\left(\hat{\boldsymbol{\delta}}_i(\boldsymbol{\alpha}),\ \hat{\boldsymbol{\delta}}_i(\boldsymbol{\beta})\right)+1}{2}\right)^p\right)\\ &= d-t-\lim_{p\to\infty}\widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta})\\ &= \lim_{p\to\infty}\frac{1}{2}\widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\alpha})+\frac{1}{2}\widetilde{\kappa}^p_\delta(\boldsymbol{\beta},\ \boldsymbol{\beta})-\widetilde{\kappa}^p_\delta(\boldsymbol{\alpha},\ \boldsymbol{\beta})\\ &= \frac{1}{2}\lim_{p\to\infty}\left\|\widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\alpha})-\widetilde{\boldsymbol{\phi}}^p_\delta(\boldsymbol{\beta})\right\|_2^2 \qquad (36)\end{aligned}$$

which completes the proof.

Acknowledgment

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for their critical and constructive comments and suggestions.

References

[1] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe, "Convexity, classification, and risk bounds," *J. Amer. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, 2006.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[4] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1914–1923.

[5] S. Cai, L. Zhang, W. Zuo, and X. Feng, "A probabilistic collaborative representation based approach for pattern classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2950–2959.

[6] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, 2011.

[7] Y. Chen, J. Mairal, and Z. Harchaoui, "Fast and robust archetypal analysis for representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1478–1485.

[8] F. Gremse, A. Höfter, L. O. Schwen, F. Kiessling, and U. Naumann, "GPU-accelerated sparse matrix–matrix multiplication by iterative row merging," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. C54–C71, 2015.

[9] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Rep. CNS-TR-2007-001, 2007.

[10] C. Guo and Q. Yang, "A neurodynamic optimization method for recovery of compressive sensed signals with globally converged solution approximating to l_0 minimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1363–1374, Jul. 2015.

[11] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, Aug. 2011.

[12] D. Huang, R. Cabral, and F. De la Torre, "Robust regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 363–375, Feb. 2016.

[13] P. J. Huber, *Robust Statistics*. Heidelberg, Germany: Springer, 2011.

[14] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM ICM*, 2014, pp. 675–678.

[15] W. Jiang, F. Nie, and H. Huang, "Robust dictionary learning with capped l_1-norm," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3590–3596.

[16] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.

[17] Z. Kang, C. Peng, and Q. Cheng, "Robust PCA via nonconvex rank approximation," in *Proc. IEEE Conf. Data Min. ICDM*, 2015, pp. 211–220.

[18] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, Dec. 2017.

[19] Z. Kang, C. Peng, and Q. Cheng, "Twin learning for similarity and clustering: A unified kernel approach," in *Proc. AAAI*, 2017, pp. 2080–2086.

[20] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl. Based Syst.*, vol. 163, pp. 510–517, Jan. 2018.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[22] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[23] H. A. Levine, "Review: A. N. Tikhonov and V. Y. Arsenin, solutions of ill posed problems," *Bull. Trans. Amer. Math. Soc.*, vol. 1, no. 3, pp. 521–524, 1979.

[24] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[25] P. Li, Q. Wang, W. Zuo, and L. Zhang, "Log-Euclidean kernels for sparse representation and dictionary learning," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1601–1608.

[26] G. Liu *et al.*, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[27] Y. Liu, F. Tang, and Z. Zeng, "Feature selection based on dependency margin," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1209–1221, Jun. 2015.

[28] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler principal component analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 498–518, 2013.

[29] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 415–422.

[30] A. Martinez and R. Benavente, "The AR face database," CVC, New Delhi, India,

[31] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, vol. 2. Philadelphia, PA, USA: SIAM, 2000.

[32] J.-X. Mi, Q. Fu, and W. Li, "Adaptive class preserving representation for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7427–7435.

[33] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint *l*2,1-norms minimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[34] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[35] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[36] W. Rudin *et al.*, *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1964.

[37] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[38] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[39] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component analysis on graphs," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2812–2820.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] Y. Tai, J. Yang, L. Luo, F. Zhang, and J. Qian, "Learning discriminative singular value decomposition representation for face recognition," *Pattern Recognit.*, vol. 50, pp. 1–16, Feb. 2016.

[42] A. Vedaldi and B. Fulkerson, "VLFeat—An open and portable library of computer vision algorithms," in *Proc. ACM ICM*, 2010, pp. 1469–1472.

[43] H. Wang, X. Lu, Z. Hu, and W. Zheng, "Fisher discriminant analysis with L1-norm," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 828–842, Jun. 2017.

[44] J. Wang *et al.*, "Robust face recognition via adaptive sparse representation," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, Dec. 2014.

[45] Y. Wang, C. Dicle, M. Sznaier, and O. Camps, "Self scaled regularized robust regression," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3261–3269.

[46] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[47] P.-Y. Wu, C.-C. Fang, J. M. Chang, and S.-Y. Kung, "Cost-effective kernel ridge regression implementation for keystroke-based active authentication system," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3916–3927, Nov. 2017.

[48] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of Gaussian kernel for one-class SVM," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 941–953, May 2015.

[49] G. Xu, B. G. Hu, and J. C. Principe, "Robust C-loss kernel classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 510–522, Mar. 2018.

[50] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2496–2504.

[51] J. Yang *et al.*, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 156–171, Jan. 2017.

[52] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 625–632.

[53] W. Yang, Z. Wang, and C. Sun, "A collaborative representation based projections method for feature extraction," *Pattern Recognit.*, vol. 48, no. 1, pp. 20–27, 2015.

[54] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[55] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 471–478.

[56] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Ann. Stat.*, vol. 32, no. 1, pp. 56–84, 2004.

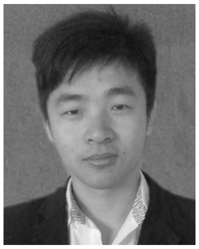
[57] Y. Zhang, B. Mu, and H. Zheng, "Link between and comparison and combination of Zhang neural network and quasi-Newton BFGS method for time-varying quadratic minimization," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 490–503, Apr. 2013.

[58] P. Zhou and J. Feng, "Outlier-robust tensor PCA," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1–9.

[59] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 42–59, 2014.

[60] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1081–1088.

[61] S. Zhu, C. Li, C.-C. Loy, and X. Tang, "Unconstrained face alignment via cascaded compositional learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3409–3417.



Shuo Chen received the B.S. degree in computer science and engineering from the Jinling Institute of Technology, Nanjing, China, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with the Nanjing University of Science and Technology, Nanjing.

His current research interests include pattern recognition, metric learning, and deep learning.

Jian Yang (M'08) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

He was a Postdoctoral Researcher with the University of Zaragoza, Zaragoza, Spain, in 2003. He was a Postdoctoral Fellow with the Biometrics Centre, Hong Kong Polytechnic University, Hong Kong, from 2004 to 2006, and the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA, from 2006 to 2007. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored over 200 scientific papers in pattern recognition and computer vision with more than 5000 Web of Science citations and 13 000 Google Scholar citations. His current research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is/was currently an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*. He is a fellow of IAPR.

Yang Wei received the B.S. degree in computer science and engineering from Nanjing University of Science and Technology, Nanjing, China, in 2015, where she is currently pursuing the Ph.D. degree in computer science and engineering.

Her current research interests include pattern recognition and computer vision, especially the graph model.

Lei Luo received the B.S. degree in fundamental mathematics from Xinyang Normal University, Xinyang, China, in 2008, and the M.S. degree in fundamental mathematics from Inner Nanchang University, Nanchang, China, in 2011. He is currently pursuing the Ph.D. degree in pattern recognition and intelligence systems with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.

His current research interests include pattern recognition and optimization algorithm.

Gui-Fu Lu received the Ph.D. degree in computer science and engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2012.

He has been teaching with the School of Computer Science and Information, Anhui Polytechnic University, Wuhu, China, since 2004. His current research interests include computer vision, digital image processing, and pattern recognition.

Chen Gong (M'16) received the B.E. degree from the East China University of Science and Technology, Shanghai, China, in 2010, the first Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University (SJTU), Shanghai, in 2016, and the second Ph.D. degree in computer science and engineering from the University of Technology Sydney, Ultimo, NSW, Australia, in 2017.

He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published over 50 technical papers at prominent journals and conferences, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, CVPR, AAAI, International Joint Conference on Artificial Intelligence (IJCAI), and ICDM. His current research interests include machine learning, data mining, and learning-based vision problems.

Dr. Gong was a recipient of the "Excellent Doctorial Dissertation" Award by SJTU and the Chinese Association for Artificial Intelligence and was nominated for the "Summit of the Six Top Talents" Program of Jiangsu Province, China, and the "Lift Program for Young Talents" of the China Association for Science and Technology. He also serves as a Reviewer for over 20 international journals such as the *Artificial Intelligence Journal*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON IMAGE PROCESSING, and is a PC Member for several top-tier conferences, such as the International Conference on Machine Learning, AAAI, International Conference on Machine Learning, ICDM, and International Conference on Artificial Intelligence and Statistics.