Self-Paced Two-dimensional PCA

Jiangxin Li,[1] **Zhao Kang,** [1*] **Chong Peng,** [2] **Wenyu Chen**[1]
[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China
[2] College of Computer Science and Technology, Qingdao University
jiangxli9898@gmail.com, zkang@uestc.edu.cn, cpeng@qdu.edu.cn, cwy@uestc.edu.cn

Abstract

Two-dimensional PCA (2DPCA) is an effective approach to reduce dimension and extract features in the image domain. Most recently developed techniques use different error measures to improve their robustness to outliers. When certain data points are overly contaminated, the existing methods are frequently incapable of filtering out and eliminating the excessively polluted ones. Moreover, natural systems have smooth dynamics, an opportunity is lost if an unsupervised objective function remains static. Unlike previous studies, we explicitly differentiate the samples to alleviate the impact of outliers and propose a novel method called Self-Paced 2DPCA (SP2DPCA) algorithm, which progresses from 'easy' to 'complex' samples. By using an alternative optimization strategy, SP2DPCA looks for optimal projection matrix and filters out outliers iteratively. Theoretical analysis demonstrates the robustness nature of our method. Extensive experiments on image reconstruction and clustering verify the superiority of our approach.

Introduction

Finding effective representation from high-dimensional data such as images and videos is a long-standing problem in the fields of machine learning, pattern recognition, and data mining (Kang et al. 2020a; Ma et al. 2020). PCA is a widely used technique for dimension reduction and feature extraction (Jolliffe 2011; Kang et al. 2020b). Numerous variants have been proposed over the past several decades (Peng et al. 2020; Liao et al. 2018; Nie, Yuan, and Huang 2014; Kang, Peng, and Cheng 2015; Gao et al. 2020). To apply them in two-dimensional data analysis, we need to convert the input matrix into one-dimensional long vector, which loses the spatial structure information embedded in pixels of image (Peng et al. 2019; Gao et al. 2019).

To leverage the inherent spatial structure, 2DPCA was proposed to directly process 2D data. Yang et al. proposed 2DPCA for face classification by directly calculating image covariance matrix (Yang et al. 2004). After that, many 2DPCA methods have been derived. For example, ℓ_1-norm is adopted to suppress outliers in 2DPCA-L1 (Li 2010; Luo et al. 2016; Ju et al. 2015), sparsity constraint is further imposed in 2DPCAL1-S (Wang and Wang 2013), nuclear-norm is also used to measure the reconstruction error (N2DPCA) (Zhang et al. 2015), a nongreedy algorithm is proposed in 2DPCA-L1-nongreedy (Wang et al. 2015), F2DPCA measures the distance in spatial dimensions with F-norm (Wang and Gao 2017). With respect to squared F-norm, an F-norm based model is more efficient in mitigating the sensitivity to outliers. Moreover, F-norm can retain the rotational invariance, a desired property of PCA. Therefore, a number of 2DPCA variants are based on F-norm (Li et al. 2017b; Wang et al. 2017). Compared to traditional PCA, 2DPCA has shown competitive performance. However, they fail to extract complicated structures and lack robust generalizable performance (Liao et al. 2018).

Recently, Zhou et al. proposed generalized centered 2DPCA with $\ell_{2,p}$-norm (G2DPCA) (Zhou et al. 2019). G2DPCA finds robust projection matrices by using the variations between each row of the projected matrix and employing the power p of the $\ell_{2,1}$-norm. It demonstrates better recognition accuracy and lower reconstruction error than many state-of-the-art 2DPCA methods, such as F2DPCA (Wang and Gao 2017), 2DPCA-L1 (Li 2010), 2DPCAL1-S (Wang and Wang 2013), N2DPCA (Zhang et al. 2015), Angle-2DPCA (Gao et al. 2018), 2DPCA-L1-nongreedy (Wang et al. 2015). However, when certain data points are overly contaminated, the existing methods are frequently incapable of filtering out and eliminating the excessively polluted ones. Furthermore, natural systems have smooth dynamics, an opportunity is lost if an unsupervised objective function remains static.

To combat aforementioned drawbacks, we propose a novel approach based on self-paced learning. Inspired by human learning, self-paced learning (Kumar, Packer, and Koller 2010) was developed to train a model from 'easy' samples to 'complex' samples. This has been shown to be beneficial in alleviating outlier issue (Jiang et al. 2018; Zhang et al. 2018), and thus improves the generalization ability. In practice, data always include both 'easy' and 'complex' samples. It would be interesting to introduce self-paced learning mechanism into 2DPCA. We explicitly model the complexity of samples and propose Self-Paced 2DPCA (SP2DPCA). A smoothed weighting scheme is utilized to dynamically evaluate the easiness of samples, so that SP2DPCA learns the clean data gradually, and simultane-

*Corresponding author.

ously prevent outliers from undermining the training process.

Related Works

For L 2D points $A_i \in \mathcal{R}^{m\times n}(i = 1, \cdots, L)$, traditional 2DPCA seeks a projection matrix $V \in \mathcal{R}^{n\times k}$ by solving (Yang et al. 2004)

$$\min_{\mathbf{V^T V=I_k}} \sum_{i=1}^{L} \|(\mathbf{A_i} - \mathbf{M}) - (\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}\|_\mathbf{F}^\mathbf{2}, \quad (1)$$

where $I_k \in \mathcal{R}^{k\times k}$ is an identity matrix, then $\mathbf{Y} = \mathbf{AV}$. Its solution corresponds to the eigenvectors of the first k largest eigenvalues of covariance matrix $\sum_i (A_i)^T (A_i)$. It can be seen that the solution is dominated by the squared large distance. Consequently, the objective function is sensitive to outlying measurements. 2DPCA-L1 proposes to use ℓ_1-norm to enhance its robustness (Wang et al. 2015; Li 2010). However, ℓ_1-norm based 2DPCA loses the rotational invariance property. To address this problem, G2DPCA adopts a general $\ell_{2,p}$-norm and preserves the structural information of data samples (Zhou et al. 2019). However, it works only in the row direction.

In (Zhang, Nie, and Li 2017a), robust 2DPCA with optimal mean (R2DPCA) is developed to alleviate outliers and project image from right and left simultaneously. This problem can be formulated as following

$$\begin{aligned} \min_{\mathbf{U,V,M}} \sum_{i=1}^{L} \big\|\mathbf{A_i} - \mathbf{M} - \mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}\big\|_\mathbf{F} \\ \mathbf{s.t.} \quad \mathbf{U^T U = I_{k_1}}, \mathbf{V^T V = I_{k_2}}, \end{aligned} \quad (2)$$

where $M \in \mathcal{R}^{m\times n}$, $U \in \mathcal{R}^{m\times k_1}$, and $V \in \mathcal{R}^{n\times k_2}$. M represents the mean matrix, U and V denote the projection matrices. Compared to other 2DPCA methods, the optimal mean M is automatically achieved, rather than traditional data preprocessing. Additionally, R2DPCA conducts a bilateral dimension reduction. To improve the robustness, F-norm is applied in (2). Capped version of (2) is also proposed to tackle the outliers that are extraordinarily huge for certain samples. In particular, if the loss is very large for certain sample, it will be replaced by a threshold ϵ, another introduced parameter. Some other methods with F-norm have also been proposed (Li et al. 2017b; Wang et al. 2017; Hu et al. 2018).

In fact, outliers have different magnitudes. Intuitively, we can assign a smaller weight to a sample with larger outliers. In such a way, we promote the robustness of model. Furthermore, human beings often learn from simple examples of learning task, then introduce complex samples step by step. This ensures us to capture the intrinsic patterns of samples with high confidence, and thus reduces the impact of outliers. This learning scheme is the so-called self-paced learning which first attempts to train a model on 'easy' samples and then gradually involves 'complex' samples (Kumar, Packer, and Koller 2010).

It has been shown that self-paced learning is more robust to hard examples like outliers and noisy points than other models (Meng, Zhao, and Jiang 2017). A number of tasks have benefited from it, such as matrix factorization (Zhao et al. 2015), multi-view clustering (Xu, Tao, and Xu 2015; Jiang et al. 2018), classification (Li et al. 2017a), multi-task learning (Li et al. 2017a). Most of these applications lie in supervised learning domain.

Most existing 2DPCA techniques intend to reduce outliers by designing different norms. They fail to explicitly differentiate difficult and easy samples. SP-PPCA (Zhao et al. 2020) eliminates the impact of outliers by introducing the self-paced learning mechanism into PPCA. However, it is not capable of directly dealing with two-dimensional data. Besides, it assigns samples binary weights instead of real-valued weights, which seems unreasonable since noise is usually non-homogeneously distributed in the data. In this paper, we seek to bridge the gap between 2DPCA and self-paced learning. Putting it differently, at a high level, we introduce self-paced learning to improve the robustness of 2DPCA.

Self-Paced 2DPCA

Inspired by the success of self-paced learning, we explicitly consider the complexity of samples and introduce human learning mechanism into 2DPCA to further enhance its robustness. In general, our framework can be written as

$$\min_{\mathbf{\Theta},\mathbf{w_i}} \sum_{i=1}^{L} \mathbf{w_i}\ell_\mathbf{i}(\mathbf{\Theta}) + f(\mathbf{w_i}, \zeta), \quad (3)$$

where $\ell_i(\Theta)$ denotes certain 2DPCA loss function with variable Θ, $\mathbf{w_i}$ is the weight for the i-th sample, and $f(\mathbf{w_i}, \zeta)$ represents the self-paced regularizer controlled by the age parameter ζ. Specifically, ζ determines the samples to be selected in each learning stage so that more examples are gradually incorporated during training.

Note that (3) is quite general for many different 2DPCA models. In this paper, we take R2DPCA as an example to demonstrate the advantage, i.e.,

$$\ell_\mathbf{i}(\mathbf{M}, \mathbf{U}, \mathbf{V}) = \big\|\mathbf{A_i} - \mathbf{M} - \mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}\big\|_\mathbf{F}.$$

Finally, our proposed Self-Paced 2DPCA (SP2DPCA) can be formulated as

$$\begin{aligned} \min_{\mathbf{U,V,M},\mathbf{w_i}} \sum_{i=1}^{L} \mathbf{w_i}\big\|\mathbf{A_i} - \mathbf{M} - \mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}\big\|_\mathbf{F} \\ + \zeta(\mathbf{w_i}\log\mathbf{w_i} - \mathbf{w_i}) \ \mathbf{s.t.}\ \mathbf{U^T U = I_{k_1}}, \mathbf{V^T V = I_{k_2}} \end{aligned} \quad (4)$$

A number of formulations for $f(\mathbf{w_i}, \zeta)$ have been designed in the literature. For simplicity, we adopt the following definition (Jiang et al. 2018)

$$f(\mathbf{w_i}, \zeta) = \zeta(\mathbf{w_i}\log\mathbf{w_i} - \mathbf{w_i}). \quad (5)$$

To simulate the human learning process, the weights of 'complex' examples should be assigned to almost zero at the beginning. As the learning process goes on, 'complex' examples are gradually included to train by assigning higher

weights. Eventually, all examples are involved in training and the weights are expected to be identical and non-zero. This is ensured by minimizing $\sum_i \mathbf{w_i} \log \mathbf{w_i}$ (i.e., maximizing the entropy). Furthermore, minimizing $-\sum_i \mathbf{w_i}$ is to avoid merely incorporating 'easy' samples by suppressing the sparsity of $\mathbf{w_i}$.

Combining Eqs.(4) and (5), we can obtain the closed-form solution for $\mathbf{w_i}$ by setting the first-order partial derivative of (4) with regard to $\mathbf{w_i}$ to zero, i.e.,

$$\mathbf{w_i^*} = e^{-\ell_i/\zeta}. \tag{6}$$

It is easy to see that $\mathbf{w_i}$ always outputs values between zero and one, so Eq.(6) assigns samples the probabilities of being 'easy'. ζ controls the speed of change of the weight w.r.t. the loss. The samples can be considered as 'easy' when its loss is less than ζ; otherwise, they are 'complex'. Hence, more 'complex' samples become 'easy' as ζ increases. Without loss of generality, we ignore the subscript i. It can been seen that, $\mathbf{w}^*$ is monotonically decreasing w.r.t. ℓ and it holds that $\lim_{\ell\to 0} \mathbf{w}^* = 1$ and $\lim_{\ell\to\infty} \mathbf{w}^* = 0$. This suggests that easy examples are preferred by the model due to their smaller losses. Additionally, $\lim_{\zeta\to 0} \mathbf{w}^* = 0$ and $\lim_{\zeta\to\infty} \mathbf{w}^* \leq 1$, i.e., $\mathbf{w}^*$ is monotonically increasing w.r.t. ζ. This indicates that more samples are included to train a mature model as ζ increases.

Optimization

By optimizing model (4), we probabilistically measure the complexity of samples. To solve this multiple variables problem, we use alternating optimization, i.e., we solve a single variable while fixing others.

Step 1: Update $\mathbf{w_i}$ for each sample as Eq.(6).

Step 2: Update $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{M}$. Following (Zhang, Nie, and Li 2017b), problem (4) can be equivalently reformulated as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{M}} & \sum_{i=1}^{L} \mathbf{d_i} \left\| \mathbf{A_i} - \mathbf{M} - \mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T} \right\|_\mathbf{F}^\mathbf{2} \\ & \mathbf{s.t.} \quad \mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{I_{k_1}}, \mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{I_{k_2}}, \end{aligned} \tag{7}$$

where $\mathbf{d_i} = \frac{\mathbf{w_i}}{\mathbf{2}\left\|\mathbf{A_i}-\mathbf{M}-\mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i}-\mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}\right\|_\mathbf{F}}$.

By setting the first-order derivative of Eq.(7) with respect to $\mathbf{M}$ to zero, we obtain

$$\mathbf{M} = \frac{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}\mathbf{A_i}}{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}} + \mathbf{U}\mathbf{N}\mathbf{V}^\mathbf{T}, \tag{8}$$

where $\mathbf{N}$ is an arbitrary constant matrix. Plugging Eq.(8) back into Eq.(7) and canceling out the redundant term, we get

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} & \sum_{i=1}^{L} \mathbf{d_i} \left\| \mathbf{A_i} - \bar{\mathbf{A}} - \mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \bar{\mathbf{A}})\mathbf{V}\mathbf{V}^\mathbf{T} \right\|_\mathbf{F}^\mathbf{2} \\ & \mathbf{s.t.} \quad \mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{I_{k_1}}, \mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{I_{k_2}}. \end{aligned} \tag{9}$$

where $\overline{\mathbf{A}} = \frac{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}\mathbf{A_i}}{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}}$. From Eq.(9), we can see that our objective function is independent of coefficient matrix $\mathbf{N}$. Thus, the optimal mean can be simplified as $\bar{\mathbf{A}}$ by setting $\mathbf{N}$ to be Null matrix.

Then, we can reformulate Eq.(7) as

$$\begin{aligned} \max_{\mathbf{U},\mathbf{V}} & \sum_{i=1}^{L} \mathbf{d_i} \mathbf{Tr}\big(\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})^\mathbf{T}\mathbf{U}\big) \\ & \mathbf{s.t.} \quad \mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{I_{k_1}}, \mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{I_{k_2}}, \end{aligned} \tag{10}$$

where $\mathbf{Tr}(\cdot)$ represents the trace of a matrix.

Denote $\mathbf{P_1} = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}(\mathbf{A_i} - \mathbf{M})\mathbf{V}\mathbf{V}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})^\mathbf{T}$, $\mathbf{P_2} = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{L}} \mathbf{d_i}(\mathbf{A_i} - \mathbf{M})^\mathbf{T}\mathbf{U}\mathbf{U}^\mathbf{T}(\mathbf{A_i} - \mathbf{M})$, problem (10) is equivalent to

$$\mathbf{U} = \underset{\mathbf{U}^\mathbf{T}\mathbf{U}=\mathbf{I_{k1}}}{\arg\max} \mathbf{Tr}(\mathbf{U}^\mathbf{T}\mathbf{P_1}\mathbf{U}) \tag{11}$$

and

$$\mathbf{V} = \underset{\mathbf{V}^\mathbf{T}\mathbf{V}=\mathbf{I_{k2}}}{\arg\max} \mathbf{Tr}(\mathbf{V}^\mathbf{T}\mathbf{P_2}\mathbf{V}). \tag{12}$$

They can be solved by the SVD of the matrix $\mathbf{P_1}$ and $\mathbf{P_2}$, respectively.

Furthermore, for a given ζ, $\mathbf{w}$ changes quickly only in a specific interval, which is very important for 'complex' samples. Therefore, we introduce a parameter c to bring the loss of each sample into the quickly changing interval for a given ζ. In addition, we divide each loss by the max loss to better control the range scaling.

$$\ell_i := \frac{c \cdot \ell_i}{\max\{\ell_1, \ell_2, \ldots, \ell_L\}}. \tag{13}$$

The impact of c is also discussed in the experiments. The complete procedures for SP2DPCA are outlined in Algorithm 1. The code of our implementation is published [1].

Theoretical Analysis

We show that the alternative optimization strategy used in algorithm 1 is actually the Majorize–Minimize (MM) scheme, which ensures its convergence. Then we analyze the property of our objective function which reveals the robustness of SP2DPCA.

Let's define $F_\zeta(\ell)$ as the integration of $\mathbf{w}^*(\ell, \zeta)$ w.r.t. ℓ:

$$F_\zeta(\ell) = \int_0^\ell \mathbf{w}^*(\ell, \zeta) d\ell = \zeta(1 - e^{\frac{\ell}{\zeta}}). \tag{14}$$

Now we consider the optimization strategy for our objective function. Our objective function (4) is hard to be analyzed since it involves three variables. In fact, SP2DPCA is minimizing a much more simplified loss where the weight $\mathbf{w}$ is eliminated.

Theorem 1. *Given fixed ζ, our proposed optimization for solving Eq.*(4) *is equivalent to the majorization-minimization algorithm (MM) for solving*

$$\min \sum_{i=1}^{L} F_\zeta(\ell_i(\mathbf{M}, \mathbf{U}, \mathbf{V})). \tag{15}$$

[1] https://github.com/sckangz/SP2DPCA.

Algorithm 1 SP2DPCA

Input:
Dataset $\mathbf{A_i} \in \mathbb{R}^{m\times n}, (i = 1, 2, \ldots, L)$;
Reduced dimension $F = k_1 \times k_2$;
Self-paced learning parameters $\zeta > 0$ and $c > 0$;
Output:
Projection matrix $\mathbf{U} \in \mathbb{R}^{m\times k_1}$, $\mathbf{V} \in \mathbb{R}^{n\times k_2}$;
Initialize $\mathbf{VV^T} = \mathbf{I_n}$, $\mathbf{UU^T} = \mathbf{I_m}$, $\mathbf{W} = \sum_{\mathbf{i}} \mathbf{A_i}/\mathbf{L}$;
while not converge **do**
  For each training sample, calculate $\ell_{\mathbf{i}}$ by Eq.() and normalize $\ell_{\mathbf{i}}$ by Eq.(13);
  For each training sample, calculate $\mathbf{w_i}$ by Eq.(6);
  while not converge **do**
    $\mathbf{d_i} = \frac{\mathbf{w_i}}{2\left\|\mathbf{A_i}-\mathbf{M}-\mathbf{UU^T}(\mathbf{A_i}-\mathbf{M})\mathbf{VV^T}\right\|_{\mathbf{F}}}$;
    $\mathbf{M} \leftarrow \frac{\sum_{\mathbf{i=1}}^{\mathbf{L}} \mathbf{d_i}\mathbf{A_i}}{\sum_{\mathbf{i=1}}^{\mathbf{L}} \mathbf{d_i}}$;
    $\mathbf{P_1} \leftarrow \sum_{\mathbf{i=1}}^{\mathbf{L}} \mathbf{d_i}(\mathbf{A_i}-\mathbf{M})\mathbf{VV^T}(\mathbf{A_i}-\mathbf{M})^{\mathbf{T}}$;
    Update the columns of $\mathbf{U}$ with the k_1 left singular vector of $\mathbf{P_1}$ corresponding to the k_1 largest singular values;
    $\mathbf{P_2} \leftarrow \sum_{\mathbf{i=1}}^{\mathbf{L}} \mathbf{d_i}(\mathbf{A_i}-\mathbf{M})^{\mathbf{T}}\mathbf{UU^T}(\mathbf{A_i}-\mathbf{M})$;
    Update the columns of $\mathbf{V}$ with the k_2 left singular vector of $\mathbf{P_2}$ corresponding to the k_2 largest singular values;
  end while
end while

Proof. Denote $Q_\zeta(\mathbf{U}, \mathbf{V}, \mathbf{M}|\mathbf{U}^*, \mathbf{V}^*, \mathbf{M}^*)$ as the first order Taylor expansion of $F_\zeta(\ell(\mathbf{U}, \mathbf{V}, \mathbf{M}))$ at $\ell(\mathbf{U}^*, \mathbf{V}^*, \mathbf{M}^*)$. For the sake of simplicity, we denote $\ell(\mathbf{U}^*, \mathbf{V}^*, \mathbf{M}^*)$ as $\ell(\mathbf{p}^*)$ and $\ell(\mathbf{U}, \mathbf{V}, \mathbf{M})$ as $\ell(\mathbf{p})$.

$$Q_\zeta(\mathbf{p}|\mathbf{p}^*) = F_\zeta(\ell(\mathbf{p}^*)) + \mathbf{w}^*(\ell(\mathbf{p}^*), \zeta)(\ell(\mathbf{p}) - \ell(\mathbf{p}^*)).$$

According to (Meng, Zhao, and Jiang 2015), it holds that

$$\begin{aligned} F_\zeta(\ell(\mathbf{p})) \le Q_\zeta(\mathbf{p}|\mathbf{p}^*) = F_\zeta(\ell(\mathbf{p}^*)) + \\ \mathbf{w}^*(\ell(\mathbf{p}), \zeta)(\ell(\mathbf{p}) - \ell(\mathbf{p}^*)). \end{aligned} \tag{16}$$

From the above equation, we know that $Q_\zeta(\mathbf{p}|\mathbf{p}^*)$ is a surrogate function of $F_\zeta(\ell(\mathbf{p}))$ for minimizing $F_\zeta(\ell(\mathbf{p}))$.

To see that the optimization for minimizing our objective function (4) is actually the MM algorithm for minimizing $\sum_{i=1}^{L} F_\zeta(\ell_i(\mathbf{p}))$, we firstly derived the following equation based on Eq. (16):

$$\begin{aligned} F_\zeta(\ell_i(\mathbf{p})) \le Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^*) = F_\zeta(\ell_i(\mathbf{p}^*)) + \\ \mathbf{w}^*(\ell_i(\mathbf{p}), \zeta)(\ell_i(\mathbf{p}) - \ell_i(\mathbf{p}^*)), \end{aligned} \tag{17}$$

where the subscription i represents the i-th sample. Next, let's look at how MM algorithm works on minimizing $\sum_{i=1}^{L} F_\zeta(\ell_i(\mathbf{p}))$ under the surrogate function $\sum_{i=1}^{L} Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^*)$.

Majorization step: Denote $\mathbf{p}^t$ as the t-th iteration learning parameter. The majorization step is to obtain each $Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^t)$ $(i = 1, 2, \ldots, L)$ by calculating each $\mathbf{w}^*(\ell_i(\mathbf{p}^t), \zeta)$:

$$\mathbf{w}^*(\ell_i(\mathbf{p}^t), \zeta) = \arg\min_{\mathbf{w_i}} \mathbf{w_i}\ell_{\mathbf{i}} + f(\mathbf{w_i}, \zeta), \tag{18}$$

which is the same as the step 1 in Section that we got Eq. (6) by fixing $\mathbf{p}^t$ in Eq. (4).

Minimization step: Then the minimization step is to update our learning parameter $\mathbf{p}^t$ based on $Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^t)$ $(i = 1, 2, \ldots, L)$ that we calculated in majorization step:

$$\begin{aligned} \mathbf{p^{t+1}} &= \arg\min_{\mathbf{p}} \sum_{i=1}^{L} Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^t) \\ &= \arg\min_{\mathbf{p}} \sum_{i=1}^{L} F_\zeta(\ell_i(\mathbf{p^t})) + \mathbf{w}^*(\ell_i(\mathbf{p^t}), \zeta)(\ell_i(\mathbf{p}) - \ell_i(\mathbf{p^t})) \\ &= \arg\min_{\mathbf{p}} \sum_{i=1}^{L} \mathbf{w}^*(\ell_i(\mathbf{p^t}), \zeta)\ell_i(\mathbf{p}). \end{aligned}$$

This is exactly the same as step 2 in Section that we got our learning parameter $\mathbf{p}$ under fixed $\mathbf{w}$ in Eq. (5).

Then it is obvious that the MM algorithm applied on $\sum_{i=1}^{L} F_\zeta(\ell_i(\mathbf{p}))$ under the surrogate function $\sum_{i=1}^{L} Q_\zeta^{(i)}(\mathbf{p}|\mathbf{p}^*)$ is equivalent to our optimization proposed to solve Eq. (5). □

Now Theorem 1 provides a new perspective on understanding our algorithm. The alternating optimization used in our algorithm is substantially the well-known MM algorithm. According to MM theory, the convergence of our algorithm is guaranteed. Moreover, with the surrogate function $F_\zeta(\ell)$, we can show that our algorithm is robust to hard samples.

Theorem 2. *Suppose that* $\min_k \ell_k > Y$ *and* $Y < \infty$*, for any pair of different samples* (i, j) *in training dataset:*

$$\left|F_\zeta(\ell_i) - F_\zeta(\ell_j)\right| \le e^{-Y/\zeta}\left|\ell_i - \ell_j\right|. \tag{19}$$

Proof. Denote $a = \min\{\ell_i, \ell_j\}$, $b = \max\{\ell_i, \ell_j\}$. From Lagrange's mean value theorem, we know that $\exists \xi \in [a, b]$, s.t.

$$F_\zeta(\ell_i) - F_\zeta(\ell_j) = \frac{\partial F_\zeta(\ell)}{\partial \ell}\Big|_{\ell=\xi}(\ell_i - \ell_j) = e^{-\xi/\zeta}(\ell_i - \ell_j).$$

Then it can be easily derived that

$$\left|F_\zeta(\ell_i) - F_\zeta(\ell_j)\right| \le \left(\sup_{\ell\in[a,b]} \left|e^{-\ell/\zeta}\right|\right)\left|\ell_i - \ell_j\right| \le e^{-Y/\zeta}\left|\ell_i - \ell_j\right|.$$

We know from Theorem 2 that compared with $\ell(\cdot)$ (i.e., the original loss), $F_\zeta(\ell(\cdot))$ is more robust toward hard instances equipped with large loss. In detail, let i be a hard instance and j an easy one. Since $e^{-Y/\zeta} < 1$, the loss difference $\left|F_\zeta(\ell_i) - F_\zeta(\ell_j)\right|$ in SP2DPCA is much smaller than the original loss difference $\left|\ell_i - \ell_j\right|$. Therefore, $F_\zeta(\ell(\cdot))$ is more robust in the sense that it is less sensitive toward large loss. □

Experiment

Experimental Setting

Our proposed method SP2DPCA is compared with a number of state-of-the-art methods, including 2DPCA (Yang et al.

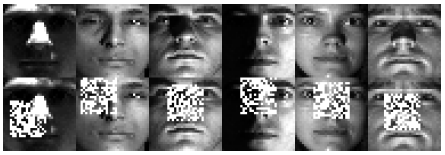


Figure 1: The first row shows some EYaleB images. The second row shows the corresponding noisy images.

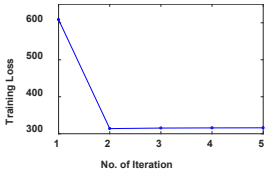


Figure 2: Convergence curve of SP2DPCA on ORL dataset.

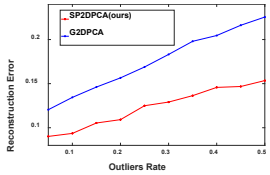


Figure 3: Mean reconstruction error versus outliers rate on ORL dataset

2004), 1-D RPCA with optimal mean (denoted by RPCA-OM) (Nie, Yuan, and Huang 2014), G2DPCA ($p = 2$) (Zhou et al. 2019), F2DPCA (Wang and Gao 2017), R2DPCA and capped R2DPCA (Zhang, Nie, and Li 2017b). We do not compare with SP-PPCA (Zhao et al. 2020) since its source code is not provided yet.

Four benchmark image databases, including ORL, MNIST, AR and EYaleB, are utilized in the experiments. For each dataset, we randomly select 20% images and place a 1/4 size occlusion. Following the comparison methods, half of the images are used for training and the rest is left for testing. Some sample images are shown in Fig.1. For SP2DPCA, the optimal parameters are searched in the range $\zeta = \{50, 100, 200, 500, 1000\}$ and $c = \{300, 500, 1000, 3000, 5000\}$ and the best results are recorded correspondingly. We stop the algorithm when the loss value does not change much.

Since 2DPCA, G2DPCA and F2DPCA perform one-side dimension reduction, we first reduce the dimension to $m \times k_2$. Next, we apply them once again on the reduced samples to obtain size $k_1 \times k_2$. For one-dimensional method RPCA-OM, we reduce the length of vector from $m \cdot n$ to $k_1 \cdot k_2$. This guarantees that all reduced data have the same dimension.

Evaluation on Reconstruction

Following previous work, we use the following reconstruction error to measure the reconstruction quality:

$$e = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x_i^o} - \mathbf{x_i^r}\|_{\mathbf{F}} \tag{20}$$

where n is the number of clean testing images, $\mathbf{x_i^o}$ is the i-th original clean image in the test set and $\mathbf{x_i^r}$ is the corresponding reconstructed image. Table 1 lists reconstruction error versus different reduced dimensions of seven methods on four databases. As it can be seen that our method obtains the best performance in all cases. In particular, SP2DPCA outperforms R2DPCA and capped R2DPCA by a large margin. Note that, our method is based on R2DPCA by introducing the self-paced learning mechanism. Our improvements over the most recent G2DPCA method are very impressive. F2DPCA also generates good performance in many cases. However, G2DPCA, F2DPCA, and R2DPCA can not produce dominanting performance as ours. This strongly verifies the benefit of adopting self-paced learning, which improves the robustness by a totally different mechanism. In many cases, RPCA-OM outperforms 2DPCA, which is because RPCA-OM computes the mean automatically and reduces the effect of outliers.

Figure 4: Reconstructed images Reconstructed images ($k_1 = k_2 = 20$) of ORL dataset. The first row shows images in the test set. The following rows show the corresponding reconstructed by SP2DPCA (Ours), R2DPCA, capped R2DPCA, F2DPCA, RPCA-OM, 2DPCA, G2DPCA, respectively.

To experimentally illustrate the robustness of our method, we show the effect of outliers rate on the reconstruction error in Fig. 3. We can see that our method is less influenced by the increase of outliers than the most recent G2DPCA method. This result indicates the robustness of our method. We also test the average running time of R2DPCA, capped R2DPCA and our method on four databases, which is 10.66s, 11.10s, 10.44s respectively. Our boost in efficiency could be attributed to the fast convergence of our algorithm. Take ORL as an example, we show that our method indeed converges quickly in Fig.2.

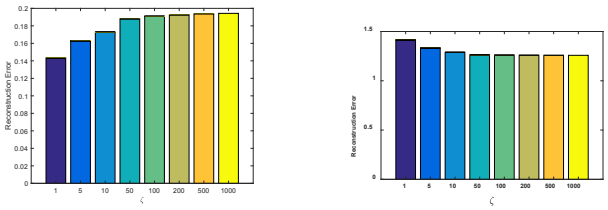
To visually see the reconstruction effect, we present some reconstructed images in Fig. 4. We can observe that the images reconstructed by our method are very close to the orig-

ORL	Dimension	14×14	15×15	16×16	17×17	18×18	19×19	20×20
	RPCA-OM	0.3005	0.2141	0.2141	0.2141	0.2141	0.2141	0.2141
	2DPCA	0.3037	0.3020	0.3010	0.2966	0.2956	0.2919	0.2905
	G2DPCA	0.1594	0.1600	0.1540	0.1455	0.1416	0.1320	0.1253
	F2DPCA	0.1684	0.1657	0.1566	0.1522	0.1419	0.1322	0.1322
	R2DPCA	0.3622	0.3668	0.3591	0.3619	0.3600	0.3651	0.3598
	capped R2DPCA	0.3056	0.4831	0.4646	0.4630	0.4674	0.4641	0.4732
	SP2DPCA(Ours)	**0.0866**	**0.0808**	**0.0765**	**0.0726**	**0.0677**	**0.0620**	**0.0572**
MNSIT	Dimension	14×14	15×15	16×16	17×17	18×18	19×19	20×20
	RPCA-OM	0.2774	0.2653	0.2514	0.2407	0.2307	0.2197	0.2088
	2DPCA	0.4757	0.4459	0.3755	0.2898	0.2568	0.2173	0.1783
	G2DPCA	0.2459	0.2240	0.2075	0.1883	0.1659	0.1508	0.1294
	F2DPCA	0.2589	0.2525	0.2368	0.2033	0.2245	0.2636	0.2220
	R2DPCA	0.2300	0.2156	0.1967	0.1790	0.1587	0.1365	0.1178
	capped R2DPCA	0.2103	0.2051	0.1918	0.1737	0.1546	0.1340	0.1226
	SP2DPCA(Ours)	**0.1753**	**0.1572**	**0.1400**	**0.1231**	**0.1083**	**0.0939**	**0.0924**
EYaleB	Dimension	14×14	15×15	16×16	17×17	18×18	19×19	20×20
	RPCA-OM	0.3640	0.3517	0.3285	0.2237	0.1798	0.1615	0.1447
	2DPCA	0.4197	0.4187	0.4183	0.4177	0.4091	0.3908	0.3683
	G2DPCA	0.2396	0.2345	0.2271	0.2217	0.2109	0.2044	0.1934
	F2DPCA	0.2439	0.2454	0.2537	0.2348	0.2509	0.2133	0.1986
	R2DPCA	0.2450	0.2417	0.2376	0.2363	0.2362	0.2356	0.2188
	capped R2DPCA	0.1936	0.2114	0.2144	0.2135	0.2113	0.2145	0.1967
	SP2DPCA(Ours)	**0.1646**	**0.1530**	**0.1430**	**0.1344**	**0.1264**	**0.1189**	**0.1111**
AR	Dimension	14×14	15×15	16×16	17×17	18×18	19×19	20×20
	RPCA-OM	0.2130	0.1876	0.1745	0.1669	0.1590	0.1438	0.1283
	2DPCA	0.3004	0.2818	0.2678	0.2625	0.2578	0.2438	0.2406
	G2DPCA	0.2271	0.2205	0.2152	0.2091	0.2019	0.1922	0.1855
	F2DPCA	0.2271	0.2239	0.2249	0.2188	0.2120	0.2056	0.2027
	R2DPCA	0.2499	0.2370	0.2312	0.2300	0.2217	0.2213	0.2184
	capped R2DPCA	0.3903	0.3623	0.3268	0.3223	0.3061	0.3072	0.3025
	SP2DPCA(Ours)	**0.1803**	**0.1693**	**0.1562**	**0.1499**	**0.1338**	**0.1240**	**0.1113**

Table 1: Reconstruction error w.r.t different reduced dimensions. The best reconstruction result under each dimension is bolded.

Dataset	Method	Clean Samples	Noised Samples
ORL	SP2DPCA with c	**0.0572**	1.2022
	SP2DPCA without c	0.2039	**1.2019**
MNIST	SP2DPCA with c	**0.0924**	0.7789
	SP2DPCA without c	0.1236	**0.7788**
EYaleB	SP2DPCA with c	**0.1111**	**1.1033**
	SP2DPCA without c	0.2039	1.1047
AR	SP2DPCA with c	**0.1113**	0.9633
	SP2DPCA without c	0.2130	**0.9623**

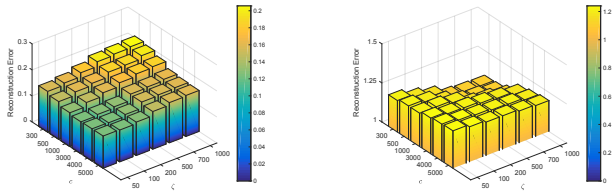
Table 2: The effect of parameter c on reconstruction error under clean and noised test samples (reduced to 20×20).



(a) Clean samples (b) Noised samples

Figure 5: The effect of ζ of SP2DPCA without c on reconstruction error under clean and noised samples of EYaleB.

inal images. Though the reconstruction error for R2DPCA, capped R2DPCA, G2DPCA, and F2DPCA in Table 1 is not so bad, their image quality is poor in most cases. This demonstrates that our method is good at preserving the spatial structure of 2D data owing to our self-paced learning mechanism. By contrast, previous methods just focus on error minimization, which might not be valid in practice. The recovered images by 2DPCA have low quality since it lacks robustness mechanism in its objective function. The reconstructed images of RPCA-OM are much worse since it loses spatial information. Some images are totally destroyed and can not be recognized. The success of SP2DPCA owns to



(a) Clean samples (b) Noised samples

Figure 6: The effect of ζ and c of SP2DPCA on reconstruction error under clean and noised samples of EYaleB.

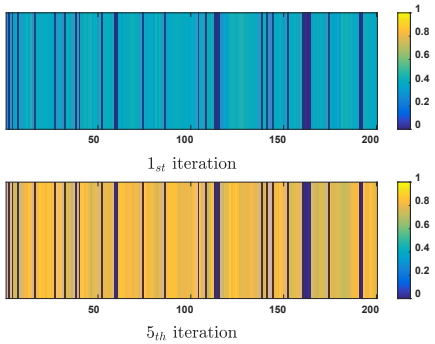
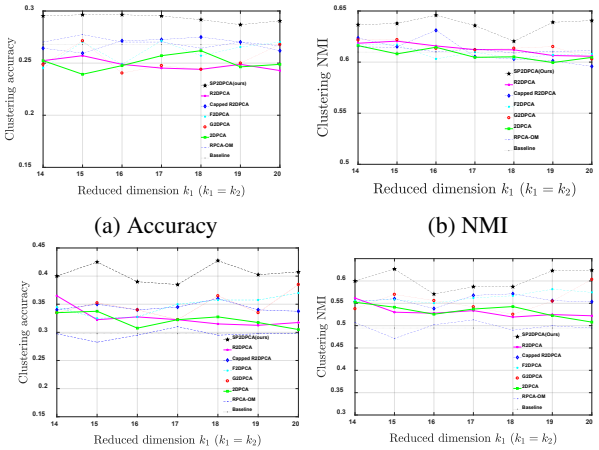


Figure 7: Visualization of the weights learned by SP2DPCA at the 1st and 5th iteration on ORL dataset.



(a) Accuracy (b) NMI

(c) Accuracy (d) NMI

Figure 8: Clustering accuracy and NMI of original images and reduced images of AR database (first row) ORL database (second row). The x-axis represents the reduced dimension k_1, while the dimension k_2 equals k_1.

it progresses from easy to hard samples, which leads to our projection vectors be less influenced by outlying images.

Parameter Analysis

To see the influence of parameter ζ and c, some experiments are designed. Table 2 lists the reconstruction error of our method with c and without c (i.e., Eq.(13) is not used.). Different from previous work, we also report the reconstruction error for noised test samples. We can see that keeping c is helpful for clean data and its influence on noisy samples is small. The reason could be that noisy items are presumably all 'complex', and reweighting is not necessary. Fig.5a presents the reconstruction error of SP2DPCA without c under different ζ values. It can be seen that a large ζ is preferred for noisy samples. In general, our performance is quite stable for a large range of ζ values. Fig.6 presents the combination effect of ζ and c. It illustrates that SP2DPCA with c has better performance when small ζ and large c are used in clean samples cases while large ζ and small c are set for noised samples.

Furthermore, we visualize the evolution of weights on ORL data. We display the weights of each sample at the 1st and 5th iteration of SP2DPCA in Fig. 7. It can be seen that weights increase as the learning process goes on. Though most weights in SP2DPCA are quite small at the beginning, they grow fast. After the 5th epoch, most samples are assigned a large weight, i.e., more 'complex' samples are involved in the learning. Moreover, a few samples still have a small weight since they are severally corrupted which should not contribute too much to our final function. This is consistent with our motivation.

Evaluation on Clustering

We further evaluate the benefit of dimension reduction by conducting clustering experiment. We perform experiments on two challenging datasets: ORL and AR which have 40 and 120 classes, respectively. After dimension reduction, k-means clustering algorithm is applied to the lower-dimensional images. Two popular metrics, Accuracy and Normalized Mutual Information (NMI), are used to evaluate the clustering performance. As a baseline, we also include the clustering results on original images.

Fig.8 presents clustering results under different reduced dimensions. The superiority of our approach can be clearly observed. In most cases, dimension reduction methods generate higher accuracy and NMI than baseline. In some cases, some techniques obtain lower accuracy and NMI than baseline, which indicates that they cause information loss during the dimension reduction process.

Conclusion

In this paper, we make the first attempt to introduce self-paced learning mechanism into 2DPCA. It is a novel way to enhance the robustness of 2DPCA. Theoretical analysis reveals the robustness nature and convergence property of our method. Extensive numerical and visual experimental results demonstrate that the proposed approach is more effective than many other state-of-the-art techniques on image reconstruction task. Experiments on clustering also show the superiority of our method on dimension reduction. Hence our proposed method is very promising for real applications.

Acknowledgements

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806045, U19A2059), the National Key R&D Program of China (Nos. 2018AAA0100204, 2018YFC0807500), the Sichuan Science and Technology Program under Project 2020YFS0057, the Ministry of Science and Technology of Sichuan Province Program (Nos. 2018GZDZX0048, 20ZDYF0343), the Fundamental Research Fund for the Central Universities under Project ZYGX2019Z015 .

References

Gao, Q.; Ma, L.; Liu, Y.; Gao, X.; and Nie, F. 2018. Angle 2DPCA: A new formulation for 2DPCA. *IEEE transactions on cybernetics* 48(5): 1672–1678.

Gao, Q.; Xu, S.; Chen, F.; Ding, C.; Gao, X.; and Li, Y. 2019. R_1 -2-DPCA and Face Recognition. *IEEE Transactions on Cybernetics* 49(4): 1212–1223. doi:10.1109/TCYB.2018.2796642.

Gao, Q.; Zhang, P.; Xia, W.; Xie, D.; Gao, X.; and Tao, D. 2020. Enhanced Tensor RPCA and Its Application. *IEEE transactions on pattern analysis and machine intelligence* .

Hu, X.; Sun, Y.; Gao, J.; Hu, Y.; and Yin, B. 2018. Locality Preserving Projection Based on F-norm. In *AAAI*.

Jiang, Y.; Yang, Z.; Xu, Q.; Cao, X.; and Huang, Q. 2018. When to Learn What: Deep Cognitive Subspace Clustering. In *MM*, 718–726. ACM.

Jolliffe, I. 2011. *Principal component analysis*. Springer.

Ju, F.; Sun, Y.; Gao, J.; Hu, Y.; and Yin, B. 2015. Image outlier detection and feature extraction via L1-norm-based 2D probabilistic PCA. *IEEE Transactions on Image Processing* 24(12): 4834–4846.

Kang, Z.; Lu, X.; Liang, J.; Bai, K.; and Xu, Z. 2020a. Relation-Guided Representation Learning. *Neural Networks* 131: 93–102.

Kang, Z.; Pan, H.; Hoi, S. C.; and Xu, Z. 2020b. Robust Graph Learning From Noisy Data. *IEEE Transactions on Cybernetics* 50(5): 1833–1843.

Kang, Z.; Peng, C.; and Cheng, Q. 2015. Robust PCA via nonconvex rank approximation. In *ICDM*, 211–220. IEEE.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *NIPS*, 1189–1197.

Li, C.; Yan, J.; Wei, F.; Dong, W.; Liu, Q.; and Zha, H. 2017a. Self-paced multi-task learning. In *AAAI*.

Li, T.; Li, M.; Gao, Q.; and Xie, D. 2017b. F-norm distance metric based robust 2DPCA and face recognition. *Neural Networks* 94: 204–211.

Li, X. 2010. L1-norm-based 2DPCA. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40(4): 1170–1175.

Liao, S.; Li, J.; Liu, Y.; Gao, Q.; and Gao, X. 2018. Robust Formulation for PCA: Avoiding Mean Calculation With L 2, p-norm Maximization. In *AAAI*.

Luo, M.; Nie, F.; Chang, X.; Yang, Y.; Hauptmann, A.; and Zheng, Q. 2016. Avoiding optimal mean robust PCA/2DPCA with non-greedy l1-norm maximization. In *IJCAI*, 1802–1808.

Ma, Z.; Kang, Z.; Luo, G.; Tian, L.; and Chen, W. 2020. Towards Clustering-friendly Representations: Subspace Clustering via Graph Filtering. In *ACM Multimedia*, 3081–3089.

Meng, D.; Zhao, Q.; and Jiang, L. 2015. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049* .

Meng, D.; Zhao, Q.; and Jiang, L. 2017. A theoretical understanding of self-paced learning. *Information Sciences* 414: 319–328.

Nie, F.; Yuan, J.; and Huang, H. 2014. Optimal mean robust principal component analysis. In *ICML*, 1062–1070.

Peng, C.; Chen, C.; Kang, Z.; Li, J.; and Cheng, Q. 2019. RES-PCA: A scalable approach to recovering low-rank matrices. In *CVPR*, 7317–7325.

Peng, C.; Chen, Y.; Kang, Z.; Chen, C.; and Cheng, Q. 2020. Robust principal component analysis: A factorization-based approach with linear complexity. *Information Sciences* 513: 581–599.

Wang, H.; and Wang, J. 2013. 2DPCA with L1-norm for simultaneously robust and sparse modelling. *Neural Networks* 46: 190–198.

Wang, Q.; and Gao, Q. 2017. Two-dimensional PCA with F-norm minimization. In *AAAI*.

Wang, Q.; Gao, Q.; Gao, X.; and Nie, F. 2017. Optimal mean two-dimensional principal component analysis with F-norm minimization. *Pattern recognition* 68: 286–294.

Wang, R.; Nie, F.; Yang, X.; Gao, F.; and Yao, M. 2015. Robust 2DPCA With Non-greedy ℓ_1-Norm Maximization for Image Analysis. *IEEE transactions on cybernetics* 45(5): 1108–1112.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view self-paced learning for clustering. In *IJCAI*.

Yang, J.; Zhang, D.; Frangi, A. F.; and Yang, J.-y. 2004. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE TPAMI* 26(1): 131–137.

Zhang, F.; Yang, J.; Qian, J.; and Xu, Y. 2015. Nuclear norm-based 2-DPCA for extracting features from images. *IEEE TNNLS* 26(10): 2247–2260.

Zhang, R.; Nie, F.; and Li, X. 2017a. Auto-weighted two-dimensional principal component analysis with robust outliers. 1, 6065–6069.

Zhang, R.; Nie, F.; and Li, X. 2017b. Auto-weighted two-dimensional principal component analysis with robust outliers. In *ICASSP*, 6065–6069. IEEE.

Zhang, Y.; Tang, Q.; Niu, L.; Dai, T.; Xiao, X.; and Xia, S.-T. 2018. Self-Paced Mixture of T Distribution Model. In *ICASSP*, 2796–2800. IEEE.

Zhao, B.; Xiao, X.; Zhang, W.; Zhang, B.; Gan, G.; and Xia, S. 2020. Self-Paced Probabilistic Principal Component Analysis for Data with Outliers. In *ICASSP 2020*, 3737–3741. IEEE.

Zhao, Q.; Meng, D.; Jiang, L.; Xie, Q.; Xu, Z.; and Hauptmann, A. G. 2015. Self-paced learning for matrix factorization. In *AAAI*.

Zhou, G.; Xu, G.; Hao, J.; Chen, S.; Xu, J.; and Zheng, X. 2019. Generalized Centered 2-D Principal Component Analysis. *IEEE transactions on cybernetics* .