Self-paced Elastic Nonnegative Matrix Factorization

Journal:	*IEEE Transactions on Neural Networks and Learning Systems*
Manuscript ID	TNNLS-2023-P-29397
Manuscript Type:	Regular Paper
Date Submitted by the Author:	04-Aug-2023
Complete List of Authors:	Mohammadi, Setare; University of Kurdistan, Computer Engineering Seyedi, Seyed Amjad; University of Kurdistan, Computer Engineering Salahian, Navid; University of Kurdistan, Computer Engineering Akhlaghian Tab, Fardin; University of Kurdistan, Computer Engineering
Keywords:	robustness, nonnegative matrix factorization, elastic loss, self-paced learning

Self-paced Elastic Nonnegative Matrix Factorization

Setare Mohammadi, Seyed Amjad Seyedi, Navid Salahian, Fardin Akhlaghian Tab

***Abstract*—Nonnegative matrix factorization (NMF) is an algebraic representation method extensively applied across various domains like data mining and machine learning. The fundamental concept behind NMF is to minimize the distance between the original input matrix and a lower-rank approximation of it. However, the basic NMF approach is not suitable for dealing with corrupted data, since its employed loss function is highly susceptible to noise and outliers. Recently, robust NMF models are introduced that choose a loss function based on the specific noise assumed on the data. However, in many real-world problems, the noise model is unknown and challenging to estimate using an ad hoc loss function. To tackle this challenge, we introduce the Self-Elastic NMF (SE-NMF) model, which adaptively combines the Frobenius norm and the $L_{2,1}$ norm. This fusion occurs within a self-paced learning (SPL) framework, where the impact of each norm is dynamically adjusted based on the learning pace. In addition, we employ a SPL soft weighting approach method to enhance the performance of the SE-NMF model. The optimization problem is solved by a proposed iterative updating algorithm. It provides efficient updating rules and incurs almost the same computational cost as conventional robust NMFs. The comprehensive experimental results on the datasets, including noisy data, demonstrate the effectiveness and robustness of Self-Elastic NMF for robust data representation.**

***Index Terms*—robustness, nonnegative matrix factorization, elastic loss, self-paced learning**

I. Introduction

DATA representation is an important topic in many real-world tasks, for example, computer vision, natural language processing, data analysis, and information retrieval [1]. It is also regarded as a key method in data analysis and machine learning. The data utilized by these domains have relatively large dimensions, so it is almost infeasible to learn directly from the original data. In this condition, an appropriate data representation method is required as a preprocessing step that projects the high-dimensional data into a low-dimensional subspace. This representation method efficiently uncovers the data underlying structures, allowing for the development of subsequent stages [2]. Matrix decomposition methods have been effectively used to create these types of data representations. Singular Value Decomposition (SVD) [3], Vector Quantization (VQ) [4], Principal Component Analysis (PCA) [5], Independent Component Analysis (ICA) [6], Concept Factorization (CF) [7], and Nonnegative Matrix Factorization (NMF) [8], [9] are some examples of matrix decomposition methods.

In the field of data representation, NMF has garnered considerable attention in recent years due to its part-based data representation. This factorization tries to decompose the input data matrix as the product of two nonnegative low-rank matrices, the basis matrix and the coefficient matrix, and achieves the closest possible approximation to the original matrix [8]. NMF has been effectively used for various data representation tasks, such as matrix completion [10], [11], data clustering [12]–[14], visual tracking [15], social network analysis [16], etc. Since encouraging the non-negativity constraint improves interpretability, a wide variety of NMF extensions have been developed in recent years, which can be classified into two divisions including, constrained NMF and generalized NMF. In the first class, some extra restrictions, such as sparsity, spatial coherence, or smoothness, are applied to the factor matrices. These restrictions are based on a side information about the desired solution, which are often task-specific, and they are applied by constraints or regularization terms in the cost function. For instance, *Ding et al.* [17], [18] introduced an improved matrix factorization model by incorporating an orthogonal constraint to the basic NMF model (i.e., ONMF), which is equivalent to K-means clustering. Furthermore, *Cai et al.* [19] introduced a Graph regularized Nonnegative Matrix Factorization (GNMF) to discover a compact representation that reveals the hidden semantics while also preserving the inherent geometric structure. The basic NMF may not always achieve enough sparsity, so it could be advantageous to explicitly regulate the level of sparsity. To address this, *Hoyer* introduced Sparse NMF, which employs a projection operator by enforcing both L_1 and L_2 norms and allows for any desired degree of sparsity [20]. CNMF introduced by *Liu et al.* [21], is a semi-supervised NMF model that leverages label information as extra hard constraints. This model enforces the consistency of coordinates in the reduced dimensional space for samples that have the same class labels.

In the second class, various objective functions that evaluate the quality of an approximation by measuring a distance between $\boldsymbol{ZH}$ and $\boldsymbol{X}$ are considered. Typically, selecting this objective function is influenced by the noise model/statistics specified for the data matrix $\boldsymbol{X}$. In addition, some NMF approaches are severely impacted by noisy data that are contaminated by outliers. The Frobenius NMF approach, which employs the least squares loss function to assess the quality of factorization, is particularly well-suited for zero-mean Gaussian noise. Nevertheless, in real-world scenarios, numerous datasets from different applications may not match the circumstances described above. According to research [22], basic NMF method is sensitive to outliers, which may greatly affect the cost function involving squared residue error [23]. To eliminate affection of noise and outliers, and to improve the robustness of NMF, numerous robust NMF techniques have been developed. One such approach is, robust matrix factorization with L_1 norm considerably mitigates the damage

* Corresponding author: Fardin Akhlaghian Tab.
Setare Mohammadi, Seyed Amjad Seyedi, Navid Salahian, and Fardin Akhlaghian Tab are with the Department of Computer Engineering, University of Kurdistan, Iran (e-mails: setare.mohammadi@uok.ac.ir; amjadseyedi@uok.ac.ir; n.salahian@uok.ac.ir; f.akhlaghian@uok.ac.ir).
Manuscript received August 4, 2023.

imposed by noise; however, it may not be able to maintain the necessary feature rotation invariance for various applications [24]. The Laplacian noise model is best supported by robust nonnegative matrix factorization using the $L_{2,1}$ norm ($L_{2,1}$-NMF) [22]. *Gao et al.* [25] introduced an effective capped $L_{2,1}$ norm NMF models entirely eliminate the influence of outliers.

In contrast to previous methods, Correntropy Induced Metric-based NMF (CIM-NMF) [26] uses correntropy instead of Squared Euclidean distance (SED) as a similarity measure. Correntropy is a nonlinear and local similarity metric used in information theoretic learning that is directly linked to the likelihood of two random variables being similar within a specific region of the joint space. The row-based CIM-NMF (rCIM-NMF) and Huber-based NMF are among the proposed variants of NMF that exhibit enhanced robustness [26]. Recently, a novel robust NMF approach has been introduced, utilizing an r-blocking logarithmic cost function denoted as $Hx(e)$. This function demonstrates greater insensitivity to noise and outlier data points compared to the Frobenius-NMF and $L_{2,1}$-NMF methods [27]. CauchyNMF method considers an isotropic Cauchy distribution to evaluate the reconstruction error, which is optimal in a maximum likelihood sense [28]. Truncated CauchyNMF tackles outliers by truncating significant errors; develop a truncated Cauchy loss to learn the subspace robustly on noise datasets contaminated with outliers [29]. More recently, an elastic loss has been proposed for NMF, namely Elastic NMF (ENMF), to be less sensitive to Gaussian and Laplacian model noises. ENMF employs a scale parameter to trade off between the Frobenius norm and the $L_{2,1}$ norm [23].

Owing to the non-convex nature of many learning models, these models simply obtain a suboptimal local solution. Recent advances in self-paced learning (SPL) [30] propose a potential solution to this local minimum issue. The key idea of SPL is to learn a model on easy instances first, and then progressively incorporate complex instances, which effectively imitates the human learning process. Various SPL realization methods for different computer vision and pattern recognition tasks have been proposed and experimentally shown to be effective for these tasks [31]–[34]. Similarly, some researches attempt to employ this learning regime in the NMF models. *Zhu and Zhang* [35] introduced MSPNMF, a NMF method that combines self-paced learning methodology with Frobenius NMF model. However, the objective function of MSPNMF employs the Frobenius norm, which is sensitive to noisy data and outliers. *Huang et al*. proposed a modified NMF method by employing the SPL method in the $L_{2,1}$-NMF so that the effect of outliers can be reduced [36].

The major drawback of self-paced learning is overfitting to a subset of samples, because of its filtering approach [37]. Each class contains both simple and complex instances. When the SPL model learns from a group of samples, it has a tendency to prefer selecting additional samples from the same class. This bias arises because those samples seem simpler to the model, given its prior knowledge. Consequently, this behavior can lead to overfitting on that specific subset, causing the model to disregard simple samples from other classes. Also, most robust and self-paced NMF models utilize an ad hoc loss function, which barely considers different noises. Elastic loss as a hybrid loss makes the Elastic NMF enable to combine the Frobenius-NMF and $L_{2,1}$-NMF. This method determines the contribution of each norm based on a predefined scale parameter. In the elastic model, a large parameter value reduces the Elastic NMF to the Frobenius-NMF, and a small parameter reduces it to the $L_{2,1}$-NMF. Therefore, setting this external scale parameter is challenging. On the other hand, to design a robust model that is insensitive to outliers, it is necessary to place more emphasis on the $L_{2,1}$ norm. In practice [23], this robust model is achieved by setting a small scale parameter; consequently, $L_{2,1}$-NMF dominates the Frobenius-NMF. Hence, this model is unable to fully utilize the capacity of the elastic mechanism.

To tackle the mentioned shortcomings and to design a novel robust NMF, this paper proposes the Self-Elastic NMF (SE-NMF) model with a complementary loss that adaptively determines the contribution of each norm (Frobenius and $L_{2,1}$) according to the learning pace. In contrast with the Elastic and existing self-paced NMF models, SE-NMF, considering both easy and complex samples, produces a curriculum that reasonably adapts the self-elastic loss to samples based on their complexity degrees. By replacing the static scale parameter of Elastic NMF [23] with the self-elastic curriculum, SE-NMF reduces its sensitivity to parameter settings and achieves a more efficient balance between the two norms. Also, SE-NMF prevents overfitting to a data subset and helps quickly grasp confident and comprehensive knowledge. In this self-paced framework, when the model is immature, the self-elastic model treats conservatively, and a large fraction of samples will be assigned to the $L_{2,1}$ loss. By gradually increasing the model maturity, the self-elastic model reduces its conservative policy, and more samples will be learned accurately by Frobenius loss. This model assigns a dynamic weight to each sample and alternately updates the weights and factor matrices by easily solving a weighted nonnegative least squares problem. In this framework, if a sample be recognized easy or noiseless, a large weight will be assigned to it, and the self-elastic error tends to the Frobenius error. In contrast, for a complex or noisy example, its elastic error tends to the $L_{2,1}$ error. By using this complementary loss function in nonnegative matrix factorization, SE-NMF introduces adaptive weights for samples with different noise models, and thus SE-NMF, in addition to being robust to noise and outliers, utilizes the benefits of self-paced learning characteristics in dealing with non-convex optimization problems. The Experimental results show that the Self-Elastic NMF is more robust than a family of robust NMF models.

The rest of this paper is structured as follows: Section 2 gives some brief background on NMF and its robust variants. In Section 3, we present the SE-NMF model in detail and derive an effective optimization algorithm to solve the proposed SE-NMF objective function. Section 4 illustrates extensive experiments on the proposed model. Finally, Section 5 concludes the paper and suggests future research.

II. Background

In this section, the basic notations and definitions are given that are important for the understanding of the paper. Also,

TABLE I
THE ELEMENT-WISE AND SAMPLE-WISE ROBUST NMF MODELS, WHERE $e_{ji} = [\boldsymbol{X} - \boldsymbol{ZH}]_{ji}$ AND $e_i = \|\boldsymbol{X} - \boldsymbol{Zh}_i\|$.

Element-wise model	**Cost function**	**Element weight** W_{ji}	**Update rules**
Frobenius-NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} e_{ji}^2$	1	$\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{(\boldsymbol{X}\odot\boldsymbol{W})\boldsymbol{H}^\top}{(\boldsymbol{ZH}\odot\boldsymbol{W})\boldsymbol{H}^\top}$ $\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{Z}^\top(\boldsymbol{X}\odot\boldsymbol{W})}{\boldsymbol{Z}^\top(\boldsymbol{ZH}\odot\boldsymbol{W})}$
L_1-NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} \lvert e_{ji}\rvert$	$(e_{ji}^2 + e^2)^{-1/2}$	
CIM-NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} (1 - \exp(-e_{ji}^2/2\sigma^2))$	$\exp(-e_{ji}^2/2\sigma^2)$	
Huber-NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} \begin{cases} e_{ji}^2 & \lvert e_{ji}\rvert \le \theta \\ 2\theta\lvert e_{ji}\rvert - \theta^2 & \lvert e_{ji}\rvert > \theta. \end{cases}$	$\begin{cases} e_{ji}^2 & \lvert e_{ji}\rvert \le \theta \\ 2\theta\lvert e_{ji}\rvert - \theta^2 & \lvert e_{ji}\rvert > \theta. \end{cases}$	
Cauchy NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} \ln(1 + e_{ji}^2/\gamma^2)$	$\frac{1}{e_{ji}^2+\gamma}$	
Truncated Cauchy NMF	$\sum_{i=1}^{n}\sum_{j=1}^{d} \begin{cases} \ln(1 + e_{ji}^2/\gamma^2) & \lvert e_{ji}\rvert \le \theta \\ \ln(1+\theta) & \lvert e_{ji}\rvert > \theta. \end{cases}$	$\begin{cases} \frac{1}{e_{ji}^2+\gamma} & \lvert e_{ji}\rvert \le \gamma\sqrt{\theta} \\ \ln(1+\theta) & \lvert e_{ji}\rvert > \gamma\sqrt{\theta}. \end{cases}$	
Sample-wise model	**Cost function**	**Sample weight** D_{ii}	**Update rules**
Frobenius-NMF	$\sum_{i=1}^{n} e_i^2$	1	$\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{XDH}^\top}{\boldsymbol{ZHDH}^\top}$ $\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{Z}^\top\boldsymbol{XD}}{\boldsymbol{Z}^\top\boldsymbol{ZHD}}$
$L_{2,1}$-NMF	$\sum_{i=1}^{n} e_i$	$1/e_i$	
rCIM-NMF	$\sum_{i=1}^{n} (1 - \exp(-e_i^2/2\sigma^2))$	$\exp(-e_i^2/2\sigma^2)$	
Hx-NMF	$\sum_{i=1}^{n} \log(1 + e_i)$	$\frac{1}{e_i(1+e_i)}$	
Self-paced NMF	$\sum_{i=1}^{n} e_i p_i$	p_i/e_i	
Elastic NMF	$\sum_{i=1}^{n} \frac{\delta e_i^2}{\delta+e_i} + \frac{e_i^2}{\delta+e_i}$	$\frac{2\delta+e_i}{2(\delta+e_i)^2}(1+\delta)$	
Self-Elastic NMF	$\sum_{i=1}^{n} e_i^2 p_i + e_i(1-p_i)$	$p_i + \frac{(1-p_i)}{e_i}$	

the different Nonnegative Matrix Factorization methods in the area of robust learning are provided.

A. Notations

This paper uses various notations to represent mathematical entities. Scalars are represented by lowercase italic letters (such as k, n, γ, etc.) and uppercase italic letters (such as G, R, Q, etc.), while vectors and matrices are denoted by boldface lowercase letters (such as $\boldsymbol{p}$, $\boldsymbol{z}$, etc.) and boldface uppercase letters (such as $\boldsymbol{Z}$, $\boldsymbol{H}$, etc.), respectively. For any matrix $\boldsymbol{M}$, its i-th column and j-th row are denoted by $\boldsymbol{m}_i$ and $\boldsymbol{m}^{(j)}$, and M_{ij} represents its $(i,\ j)$-element. The trace of $\boldsymbol{M}$ is represented by $\mathrm{Tr}(\boldsymbol{M})$, and the transposed matrix of $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^\top$. The paper introduces the Frobenius norm of a matrix $\boldsymbol{M} \in \mathbb{R}^{d\times n}$ as $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} M_{ji}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{M}^\top\boldsymbol{M})} = \sqrt{\mathrm{Tr}(\boldsymbol{M}\boldsymbol{M}^\top)}$. Additionally, the $\ell_{2,1}$-norm for $\boldsymbol{M}$ is defined as $\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^{n}\|\boldsymbol{m}_i\| = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{d} M_{ji}^2}$.

B. Nonnegative Matrix Factorization

Given a d dimensional vector $\boldsymbol{x}$ with nonnegative entries, whose n observations are denoted as $\boldsymbol{x}_i, i = 1, 2, \ldots, n$, let the input matrix be $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}_{\ge 0}^{d\times n}$, NMF seeks to factorize $\boldsymbol{X}$ into nonnegative basis $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_r] \in \mathbb{R}_{\ge 0}^{d\times r}$ and nonnegative coefficient matrices $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n] \in \mathbb{R}_{\ge 0}^{r\times n}$, such that $\boldsymbol{X} \approx \boldsymbol{ZH}$ [8].

Given the observation $\boldsymbol{x}_i \in \mathbb{R}_{\ge 0}^{d}$ (ith data point), NMF decomposes it into basis $\boldsymbol{Z} \in \mathbb{R}_{\ge 0}^{d\times r}$ and the representation $\boldsymbol{h}_i \in \mathbb{R}_{\ge 0}^{r}$, i.e.,

$$\min_{\boldsymbol{Z},\boldsymbol{H}\ge 0} \sum_{i=1}^{n} \ell_i(\boldsymbol{x}_i, \boldsymbol{Zh}_i), \tag{1}$$

where $\ell(.)$ is a loss function. It is obvious that $\boldsymbol{h}_i$ is the weight coefficient of the seen entries $\boldsymbol{x}_i$ on the columns of $\boldsymbol{Z}$, the latent vectors of $\boldsymbol{X}$. Therefore, NMF factors each data into the linear or nonlinear combination of the base vectors. Because of the precondition $r \ll min(d, n)$, the obtained base vectors are incomplete over the original space. In other words, this approach attempts to represent a high-dimensional pattern with far fewer bases, so the perfect approximation can be achieved successfully only if the intrinsic features are identified in $\boldsymbol{Z}$.

There are numerous types of losses that are commonly used in NMF, including element-wise and sample-wise NMF approaches. Robust element-wise NMF approach focuses on decomposing individual elements of the input matrix rather than the entire matrix as a whole. By focusing on individual elements, it provides a fine-grained analysis of the data. Table I showcases some well-known models. However, if your data exhibits sample-specific characteristics, you can adopt a sample-wise NMF model. This involves applying NMF separately to each sample, allowing for a more customized factorization. Sample-wise NMF can capture sample-specific patterns and improve the interpretability of the extracted features [38]. Table I shows a selection of well-known sample-wise methods. The choice between these two approaches depends on the specific characteristics of the data and the objectives of the analysis.

1) Frobenius-NMF: The Frobenius-NMF [9] considers that the noise follows a Gaussian distribution and derives the objective function based on the squared L_2 norm as follow:

$$C_F = \min_{\boldsymbol{Z},\boldsymbol{H}\ge 0} \|\boldsymbol{X} - \boldsymbol{ZH}\|_F^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Zh}_i\|^2. \tag{2}$$

The multiplicative update rule (MUR) is a well-known method for solving NMF. Owing to the outstanding mathematical properties of the squared L_2 norm and the effectiveness of MUR, NMF has been adapted for many different problems [19], [39], [40]. However, Frobenius-NMF and its modifications are non-robust since the L_2 norm is sensitive to outliers. Figure 1 (a)

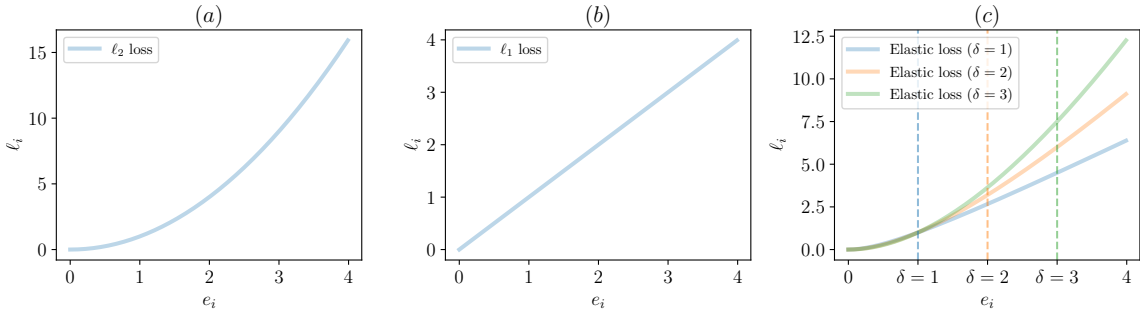


Fig. 1. Different loss functions for Robust NMFs. *a*) Frobenius NMF loss function; *b*) $L_{2,1}$-NMF loss function; *c*) Elastic NMF loss function with 3 different predefined parameters.

plots the curves of this objective function.

2) $L_{2,1}$-NMF: Since NMF is significantly a sum of the squared L_2 norm of the errors, the larger errors control the cost function and make NMF non-robust. To deal with this issue, *Kong et al*. [22] proposed NMF based on the $L_{2,1}$ norm, which optimizes the $L_{2,1}$ norm of the error matrix. Contrary to NMF, $L_{2,1}$-NMF is more robust due to the fact that the influence of outliers is inhibited during subspace learning. $L_{2,1}$-NMF enables effective and elegant updating rules and faces nearly the same computational cost as Frobenius-NMF, hence becoming a candidate to be used in many real-world tasks.

$$C_{2,1} = \min_{\boldsymbol{Z},\boldsymbol{H}\geq 0} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H}\|_{2,1} = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|. \quad (3)$$

In Figure 1 (b), the objective function is depicted.

3) Elastic NMF: Frobenius-NMF and $L_{2,1}$-NMF, evaluate the quality of factorization based on the assumption of independent and identically distributed Gaussian noise model and Laplacian noise model, respectively. *Xiong et al.* [23] introduced an elastic loss that combines the Frobenius norm and the $L_{2,1}$ norm, allowing for adaptability when the noise distribution is uncertain. This enhances the robustness of ENMF against noise and outliers. The elastic loss is utilized to optimize matrix factorization and is defined as follows:

$$C_{el} = \min_{\boldsymbol{Z},\boldsymbol{H}\geq 0} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2 + \sum_{i=1}^{n} \beta_i \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|, \quad (4)$$

where β_i is the loss parameter. The optimal solution of Equation (4) could be mathematically different from its equivalent versions of Frobenius-NMF and $L_{2,1}$-NMF. The contribution of each norm for each sample is highly depended to the predefined loss parameter β_i. The asymptotic property of this function is,

$$\begin{aligned} &\text{As } \beta_i \to 0, \text{then } C_{el}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i) \to C_F(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i), \quad (5)\\ &\text{As } \beta_i \to \infty, \text{then } C_{el}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i) \to C_{2,1}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i). \end{aligned}$$

Since all β_i values are not related to the scales of residues, Equation (4) is defined as a hard elastic loss. The primary drawback of the hard elastic loss presented in Eq. (4) lies in the necessity of appropriately defining the value of β_i. To cover this drawback, β_i should be considerably associated with the residue e_i. Considering the aforementioned definition, the loss parameter β_i is set as follows:

$$\beta_i = \frac{e_i - e_i^2}{e_i + \delta} \quad (6)$$

where $e_i = \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|$. By substituting (6) into (4), we have a closed form of soft elastic loss defined as $C_{el}(.)$:

$$\begin{aligned} C_{el} &= \min_{\boldsymbol{Z},\boldsymbol{H}\geq 0} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H}\|_{el} \quad (7)\\ &= \min_{\boldsymbol{Z},\boldsymbol{H}\geq 0} \underbrace{\sum_{i=1}^{n} \frac{\delta \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2}{\delta + \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|}}_{\ell_2 \text{ pseudo-loss}} + \underbrace{\sum_{i=1}^{n} \frac{\|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2}{\delta + \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|}}_{\ell_1 \text{ pseudo-loss}} \end{aligned}$$

where the first term is referred to as L_2 pseudo loss and the second term is regarded as L_1 pseudo loss, as indicated by the scale parameter δ. Each loss's contribution is determined by the scaling component δ and the residue value e_i. The soft elastic NMF has the following properties:

- As $e_i \gg \delta$, then $C_{el}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i) \propto C_{2,1}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i)$
- As $e_i \ll \delta$, then $C_{el}(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i) \propto C_F(\boldsymbol{x}_i, \boldsymbol{Z}\boldsymbol{h}_i)$

Figure 1 (c) plots the curves of this objective function with different scale parameter δ.

III. Proposed Model

This section proposes the Self-Elastic Nonnegative Matrix Factorization (SE-NMF), which integrates both Frobenius and $L_{2,1}$ factorization model for learning a robust data representation in a self-paced learning paradigm. The fundamental reasons for its success are listed as follows: (1) It combines the two widespread losses to obtain a more generalized latent representation, which also provides a natural solution for robust unsupervised learning; (2) It utilizes a modified self-paced learning method, and so the factorization can learn easy and complex samples by cautiously adopting its cost function; (3) It automatically adopts an appropriate combined loss in a dynamic sample-wise weighting scheme without the requirement for accurate manual specification of the trade-off parameter; (4) It unifies the above into one joint learning problem and employs an effective alternating optimization method.

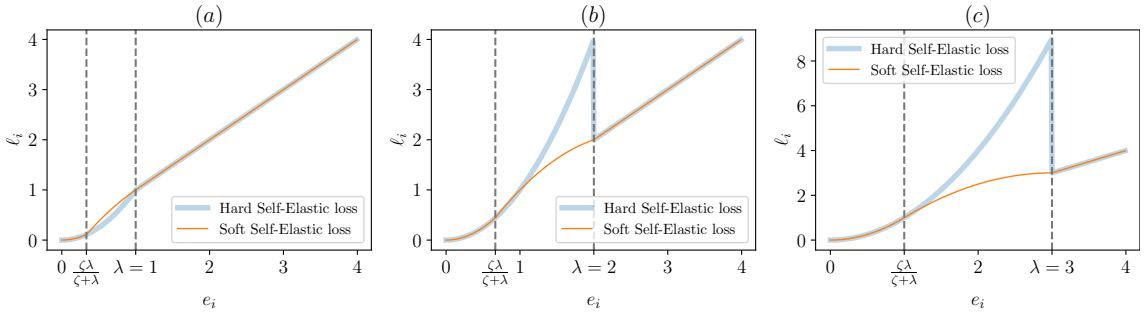


Fig. 2. Three stages of the proposed Self-paced Elastic NMF loss function with the growth of the complexity threshold.

A. Basic model

The conventional robust strategy is based on a worst-case analysis and may lead to an overly conservative policy. Therefore, most existing robust loss functions suffer from an underfitting problem, and are not sufficient by themselves to train accurate models [41]. On the contrary, squared error reconstruction functions increase the error by square, which can lead to fast convergence but also increase the risk of overfitting. However, if the residual of an outlier sample surpasses the residuals of other regular samples, it can easily dominate the entire objective function. Therefore, the outlier samples have a significant influence on the learning model efficiency, potentially leading to inaccurate results. These problems can be handled in a selective manner by fusing non-robust and robust loss functions properly.

In this paper, we address the dilemma between overfitting and underfitting in learning from noisy samples with complementary loss functions. Different from previous methods, this method uses an optimizable weight vector to measure the complexity of the samples in the matrix factorization framework. In our model, both non-robust Frobenius and robust $L_{2,1}$ cost functions are incorporated, which complement each other in a joint learning objective. To design an elastic loss with a more efficient combination of losses, we assign a weight to each sample, which is used for re-weighting the loss of the sample in the NMF cost function. This weighting is equivalent to altering the data distribution and thus modifying the factorization. More specifically, a weight $0 \leq p_i \leq 1$ is assigned to sample $\boldsymbol{x}_i$ for training in the Frobenius-NMF, and then a complementary weight $1 - p_i$ to $\boldsymbol{x}_i$ for training in the $L_{2,1}$-NMF. Finally, re-weighting $\boldsymbol{p}$ during the evolution of learning results in a more general model as follows:

$$\min_{\boldsymbol{Z},\boldsymbol{H},\boldsymbol{p}\geq 0} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2 p_i + \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|(1 - p_i) \quad (8)$$

In the objective function (8), $\boldsymbol{p}$ can be solved in an alternative minimization algorithm. In addition, $\boldsymbol{p}$ is quite simple to solve when $\boldsymbol{Z}$ and $\boldsymbol{H}$ are fixed. In this framework, the model is able to filter out a proportion of difficult samples and learns the reliable easy samples using Frobenius loss and the hard samples using $\ell_{2,1}$ loss. As a result, the model can capture a more accurate subspace and reduce the effect of noise on the model to a quite low level. The proposed loss function is illustrated in Figure 2.

B. Self-paced Learning

We briefly describe the original self-paced learning (SPL) framework as bridging knowledge before describing our novel approach. SPL aims to learn a weight variable $\boldsymbol{p} = [p_1, \ldots, p_n]^\top$ as well as the model parameter Θ. The original objective function of SPL is as follows [30]:

$$\min_{\Theta,\boldsymbol{p}} \sum_{i=1}^{n} p_i \ell_i(\boldsymbol{x_i}, \Theta) + f(\lambda, \boldsymbol{p}), \quad (9)$$

where $\ell_i(\boldsymbol{x}_i, \Theta)$ stands for the approximation error of the i-th sample as determined by a loss function, λ is the pace parameter, and n stand for the number of samples. Traditional SPLs generally force $\boldsymbol{p} \in \{0, 1\}^n$ and determinate $f(\lambda, \boldsymbol{p})$ as:

$$f(\lambda, \boldsymbol{p}) = -\lambda \sum_{i=1}^{n} p_i, \quad (10)$$

and, the optimum values of $\boldsymbol{p}$ is defined by:

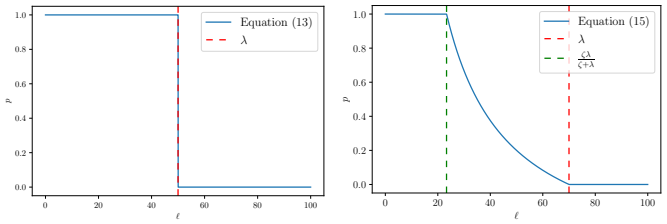
$$p_i^* = \begin{cases} 1, & \text{if } \ell_i < \lambda, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

As it can be seen from Figure 3 (a), λ is utilized to determine which samples are chosen for training. The cost function minimization procedure prefers to choose samples with small loss values when the value of λ is small. As the number of iterations increases, the method steadily increases the value of λ. As a result, the SPL process may be expressed as commencing with a small number of easy samples and gradually selecting more examples until all of the samples are chosen to train the model.

C. Hard Self-Elastic NMF

To introduce the proposed self-elastic model, we modify the self-paced learning framework (9) to adapt it to the complementary loss function (8). We propose a hard self-elastic model by minimizing the following objective function:

$$\min_{\boldsymbol{Z},\boldsymbol{H},\boldsymbol{p}\geq 0} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2 p_i + \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|(1 - p_i) \quad (12)$$
$$- \lambda \sum_{i=1}^{n} p_i,$$



(a) Hard self-paced function (b) Soft self-paced function

Fig. 3. Self-paced regularization functions.

and optimal solution for $\boldsymbol{p}$ is

$$p_i^* = \begin{cases} 1, & \text{if } \ell_i < \lambda \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\ell_i = e_i^2 - e_i$ is derived from differentiating (12) with respect to p_i.

The proposed model is applied through an iterative updating scheme and a thresholding strategy. At the beginning of training, by assuming the model is immature, the self-paced regime starts with a very small threshold λ. Samples with a loss smaller than λ are considered easy samples, while those with a loss larger than λ are deemed complex samples. These easy and complex samples are selected for learning using Frobenius-NMF and $L_{2,1}$-NMF, respectively. As the threshold grows (i.e., in the subsequent self-paced iterations), the risk of error gradually decreases, and more samples with larger square losses will be gradually assigned to Frobenius-NMF, and the more samples contribute to learning the subspace. This approach is equivalent to adapting the scale parameter in the Elastic NMF [23] according to the learning pace, so that, a small parameter allocates more contribution to $L_{2,1}$-NMF at the beginning, and by increasing this parameter gradually, the contribution will be transferred to the Frobenius-NMF.

D. Soft Self-Elastic NMF

The hard self-elastic model assigns binary weights $\boldsymbol{p}$ to the samples certainly, which is an imperfect approach for many real applications, and it has been demonstrated that soft weighting is more effective than the hard way [42]. Furthermore, the noise embedded in the data is usually non-homogeneous across samples. Soft weighting, which assigns continuous real-valued weights, more accurately captures the underlying significance of samples during the learning process.

To meet this drawback, we employ a soft self-paced learning method that modifies the NMF loss function to produce continuous weights that lie within the range of [0, 1] instead of being restricted to binary values, and uses a logarithmic function to determine the uncertainty of weights [43] (See Figure 3 (b)). Rather than using hard weighting, the proposed formulation enables us to create a soft regularization as follows:

$$g(\lambda, \zeta, \boldsymbol{p}) = -\sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda), \tag{14}$$

and optimal solution for $\boldsymbol{p}$ is

$$p_i^* = \begin{cases} 1, & \text{if } \ell_i \leq \zeta\lambda/(\zeta+\lambda) \\ 0, & \text{if } \ell_i > \lambda \\ \zeta/\ell_i - \zeta/\lambda, & \text{otherwise} \end{cases} \tag{15}$$

where $\ell_i = e_i^2 - e_i$. It is obvious that (15) uses a soft weighting mechanism. Additionally, we set $\zeta{=}\frac{\lambda}{2}$ for experimental simplicity. By adding (14) to the basic model (8), we have a closed form of soft SE-NMF model defined as:

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{H},\boldsymbol{p}\geq 0} & \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2 p_i + \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|(1-p_i) \\ & - \sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda), \end{aligned} \tag{16}$$

In this robust framework, a self-paced approach is utilized to fuse the Frobenius and $L_{2,1}$ factorizations in the initial and secondary expressions. Meanwhile, the regularization term adjusts the impact of each factorization.

E. Numerical Solution

The non-convex problem (16) can be addressed using alternating minimization (Algorithm 1), which allows us to update the variables iteratively until we reach a satisfactory solution. We use gradient descent to update one of the variables by fixing the others in each iteration. As a result, the entire optimization problem is then broken down into a series of smaller, easier-to-solve subproblems. To solve the objective function (16) in a weighted NMF framework, it can be rewritten as

$$\min_{\boldsymbol{Z},\boldsymbol{H},\boldsymbol{p}\geq 0} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|^2 d_i - \sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda), \tag{17}$$

where $d_i = p_i + (1-p_i)/\|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|$. The objective function (17) can be rewritten in the trace form to be solved by Multiplicative Update Rule (MUR) method [44] as follows:

$$\begin{aligned} C_{sp} =& \mathrm{Tr}[(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H})\boldsymbol{D}(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H})^\top] - \sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda) \\ =& \mathrm{Tr}[\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top - 2\boldsymbol{X}\boldsymbol{D}\boldsymbol{H}^\top\boldsymbol{Z}^\top + \boldsymbol{Z}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top\boldsymbol{Z}^\top] \\ & - \sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda) \end{aligned} \tag{18}$$

where $D_{i,i} = p_i + (1-p_i)/\|\boldsymbol{x}_i - \boldsymbol{Z}\boldsymbol{h}_i\|$. To solve the function (18) with nonnegativity constraint, we define Lagrangian multipliers $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ that force the nonnegativity on $\boldsymbol{Z}$ and $\boldsymbol{H}$, respectively.

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{H}} & \mathrm{Tr}[-2\boldsymbol{X}\boldsymbol{D}\boldsymbol{H}^\top\boldsymbol{Z}^\top + \boldsymbol{Z}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top\boldsymbol{Z}^\top] \\ & - \mathrm{Tr}(\boldsymbol{Z}^\top\boldsymbol{\Phi}) - \mathrm{Tr}(\boldsymbol{H}^\top\boldsymbol{\Psi}) - \sum_{i=1}^{n} \zeta \ln(p_i + \zeta/\lambda) \end{aligned} \tag{19}$$

By fixing all the variables (except $\boldsymbol{Z}$), and setting the partial derivative of (19) with respect to $\boldsymbol{Z}$ to $\boldsymbol{0}$, we obtain:

$$\boldsymbol{\Phi} = -2\boldsymbol{X}\boldsymbol{D}\boldsymbol{H}^\top + 2\boldsymbol{Z}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top \tag{20}$$

Algorithm 1 Self-Elastic NMF (SE-NMF)

Input: data matrix $\boldsymbol{X}$, latent factor k, pace parameter λ, scale parameter ζ, and growth parameter μ;
Output: $\boldsymbol{W}, \boldsymbol{H}$;

while convergence not reached **do**
  Update self-paced weights $\boldsymbol{p}$ according to (13) (Hard SE-NMF) or according to (15) (Soft SE-NMF);
  $D_{i,i} \leftarrow (p_i + (1 - p_i)/e_i)$;
  $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot (\boldsymbol{XDH}^\top / \boldsymbol{ZHDH}^\top)$;
  $\boldsymbol{H} \leftarrow \boldsymbol{H} \odot (\boldsymbol{Z}^\top \boldsymbol{XD} / \boldsymbol{Z}^\top \boldsymbol{ZHD})$;
  $\lambda \leftarrow \mu\lambda$
end while
return $\boldsymbol{W}, \boldsymbol{H}$;

With respect to the complementary slackness condition of the Karush-Kuhn-Tucker [44], we have:

$$\boldsymbol{\Phi} \odot \boldsymbol{Z} = \boldsymbol{0}, \tag{21}$$

where $\odot$ shows the Hadamard product. This equation is the fixed point that the solution must guarantee convergence. By solving (21), we have the following updating rule:

$$\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{XDH}^\top}{\boldsymbol{ZHDH}^\top} \tag{22}$$

Similarly, by fixing all the variables except for $\boldsymbol{H}$, and setting the partial derivative of (19) with respect to $\boldsymbol{H}$ to $\boldsymbol{0}$, we obtain:

$$\boldsymbol{\Psi} = -2\boldsymbol{Z}^\top \boldsymbol{XD} + 2\boldsymbol{Z}^\top \boldsymbol{ZHD} \tag{23}$$

The updating rule for $\boldsymbol{H}$ is obtained, following like derivation of the update rule for $\boldsymbol{Z}$,

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{Z}^\top \boldsymbol{XD}}{\boldsymbol{Z}^\top \boldsymbol{ZHD}} \tag{24}$$

Similarly, the weight variable p_i in either the hard model or the soft model will be updated iteratively according to (13) and (15), respectively. The optimization process of SE-NMF is provided in Algorithm 1.

IV. Experimental

In this section, we investigate and prove the robustness and effectiveness of the proposed model by comparing with 11 basic and state-of-the-art models on five benchmark datasets. In addition to the experimental settings, results analysis, parameter study, and convergence analysis are discussed in detail. For all of the compared NMF-based algorithms, to eliminate the effect of initial values, we run each method five times with different various initializations and the average results and their standard deviations are reported. Also, the multiplicative updating rules for factor matrices are executed 700 iterations. We determined the parameters for each compared algorithm according to original papers where the method were first introduced. The number of latent components is defined to be equal to the number of clusters in each dataset, and we employ the original k-means clustering method on the representation matrix $\boldsymbol{H}$ for evaluating the clustering performance. Three frequently used evaluation criteria are used to evaluate the performance of clustering, including Accuracy (ACC) [45], Normalized Mutual Information (NMI) [46], and Adjusted Rand Index (ARI) [47].

TABLE II
Details of real-world datasets

Dataset	#sample	#feature	#class	Application
Yale	165	1024	15	Face recognition
ORL	400	1024	40	Face recognition
COIL20	1440	1024	20	Object recognition
MNIST	1000	784	10	Handwriting recognition
Fashion-MNIST	1000	784	10	Cloth recognition

A. Datasets

To evaluate and compare the performance of the proposed data representation model, we perform experiments with different noise models, including Gaussian, Laplacian, and occluded noises on five image datasets which have been extensively employed to evaluate the efficacy of matrix factorization models. Details of the datasets are described in the following section and summarized in Table II.

- **Yale:** The Yale face dataset, originally introduced by *Belhumeur et al.* [48] in 1997, comprises 15 individuals, with each person photographed in 11 different images under varying conditions, including center-light, with/without glasses, happy, left-light, normal, right-light, sad, sleepy, surprised, and wink. In our experiments, each image is scaled to 32×32 pixels.
- **ORL:** This dataset consists of 400 images of 40 separate people, each person has 10 images at different times with different light intensity and facial expressions. In our test, the dimensions of each image are resized to 32×32 pixels [49].
- **COIL20:** The object image dataset consists of 20 distinct objects that have been photographed from all angles in a 360-degree rotation. In our experiment, we resized each image to 32×32 pixels [50].
- **MNIST:** The original dataset contains 70,000 digital images of handwritten digits, each with a resolution of 28x28 pixels. To create a smaller dataset, we randomly selected 100 images for each digit (ranging from "0" to "9") [51].
- **Fashion-MNIST:** Fashion-MNIST is designed to be a substitute for the original MNIST dataset and is utilized as a standard for evaluating various machine learning techniques [52]. This dataset comprises of 60,000 training set images and 10,000 test set images. Each image is grayscale, with dimensions of 28×28 pixels, representing a sample and is associated with one of 10 categories. We choose 100 random samples from each class to create a new dataset with 1000 digit images.

B. Noisy datasets

For the purpose of verifying the robustness of SE-NMF, we generate the noisy datasets by adding random Gaussian, Laplacian, and occlusion noises to each pixel of five image datasets. Gaussian noise is one of the common and most

(a) Clean dataset (b) Gaussian Noise (c) Laplacian Noise (d) Occlusion

Fig. 4. The Illustration of datasets with different types of noise

prevailing types of random noise that follows a normal distribution and it can be added to a dataset to simulate randomness or variability in the data. To simulate this type of noise, we choose parameters of Gaussian distribution with mean 0 and sigma 0.4 because the maximum pixel value is 1. In these process, 40 percent of all pixels in each image has been corrupted for all datasets. Figure 4 (b) demonstrated simulated Gaussian noise on the Yale dataset with mentioned parameters above. Laplacian noise exists in many types of data, e.g., gradient-based image features such as SIFT [53]. In this paper, the Laplacian noise is generated with mean 0 and deviation 0.1. In figure 4 (c), the produced Laplacian noise version on the Yale dataset has been illustrated. Image occlusion is a prevalent type of noise that can degrade image recognition performance. Different to the diffuse noise of the Gaussian and Laplacian distributions, block occlusion obscures all details from a single distinct area. In this experiment, we utilize contiguous occlusion to generate occluded noise version. Specifically, we randomly position a 12×12-sized block on each sample within all datasets and then set the pixels within that block to zero. Figure 4 (d) gives examples of this type of simulated noise.

C. Evaluation metrics

In order to evaluate the clustering performance, we present three commonly used quantitative measures in this section: Normalized Mutual Information (NMI), Accuracy (ACC), and Adjust Rand Index (ARI). The details of these evaluation metrics are provided as follows:

- **Normalized mutual information (NMI):** Is a method utilized to evaluate the quality of clustering from the information theory perspective. It is computed by normalizing the mutual information between the cluster assignments and the pre-existing input labeling of the classes. The normalization is accomplished by averaging the entropy of the cluster assignment and that of the pre-existing input labeling. Formally, let $\boldsymbol{y}$ represent the pre-existing classes, $\boldsymbol{c}$ be a specific clustering outcome, $MI(\boldsymbol{c}, \boldsymbol{y})$ is the mutual information of clustering assignment with pre-existing class labels, and $H(\boldsymbol{c})$ is the entropy for the clustering assignment. A higher NMI value is indicative of a better clustering solution [46]. The NMI is defined as:

$$NMI(\boldsymbol{c}, \boldsymbol{y}) = \frac{MI(\boldsymbol{c}, \boldsymbol{y})}{max(H(\boldsymbol{c}), H(\boldsymbol{y}))}. \tag{25}$$

- **Clustering accuracy (ACC):** This metric is used to assess clustering performance, which establishes a one-to-one correspondence between clusters and classes. For a given point x_i, c_i and y_i variables denote the clustering outcome and the true label, respectively. The ACC is defined as:

$$ACC(\boldsymbol{c}, \boldsymbol{y}) = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n}, \tag{26}$$

The $map(.)$ function corresponds to the optimal mapping function, and $\delta(x, y)$ is the delta function, with $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise. The mapping function $map(.)$ aligns the true class label with the obtained clustering label, and the optimal mapping is determined using the Hungarian algorithm [45]. A high ACC value suggests a superior clustering performance.

- **Adjusted Rand Index (ARI):** Is a metric that assesses the resemblance between two data clusters. ARI has a negative value when the agreement between two classes is lower than what would be expected by chance, and a value of 1 when the clusters are identical. The formula for ARI is as follows:

$$ARI(\boldsymbol{c}, \boldsymbol{y}) = \frac{a - b.d}{1/2[b + c] - b.d}, \tag{27}$$

where $a = \sum_{i,j} \binom{n_{ij}}{2}$, $b = \sum_i \binom{n_{i.}}{2}$, and $c = \sum_j \binom{n_{.j}}{2}$, and $d = \sum_j \binom{n_{.j}}{2} / \binom{n}{2}$. In the given equation, c refers to the clustering outcomes, and y represents the ground truth clustering labels. n_{ij} denotes the number of identical samples in both cluster c_i and cluster y_j ,while $n_{i.}$ and $n_{.j}$ refer to the number of identical samples in cluster c_i and cluster y_j, respectively. It is important to note that the Adjusted Rand Index (ARI) is a chance-adjusted variant of the Rand Index (RI) utilized as an external criterion for comparing clustering outcomes. While RI has a range of $[0\ \ 1]$, ARI has a range of $[-1,\ 1]$ [47].

D. Compared methods

To verify the superior performance of the SE-NMF model for data representation, we consider the other 11 Robust NMF models as the comparison methods, which are listed as follows:

- **Frobenius-NMF** [8]: The conventional NMF minimizes a predetermined square error loss, and is regarded as a baseline model.
- $\boldsymbol{L_1}$**-NMF** [24]: Element-wise NMF which measures the loss by using the L_1 norm of the error matrix.
- $\boldsymbol{L_{2,1}}$**-NMF** [22]: A robust formulation of NMF that uses $L_{2,1}$ norm based loss to replace the least square loss function of the Frobenius-NMF.
- **Capped norm NMF** [25]: Robust Capped norm NMF (RCNMF) filters out the effect of outlier samples by limiting their proportions in its $L_{2,1}$ objective function.
- **CIM-NMF** [26]: A sample-wise NMF model based on the Correntropy Induced Metric function.
- **Huber-NMF** [26]: An element-wise NMF model with the Huber function to calculate the approximation errors.

TABLE III
THE COMPARISON RESULTS ON THE YALE DATASET FOR DIFFERENT TYPES OF NOISE, EVALUATED BASED ON NMI, ACC, AND ARI METRICS.

methods		Clean			Gaussian			Laplacian			Occluded		
		NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
NMF	mean	0.494	0.462	0.233	0.472	0.438	0.209	0.474	0.441	0.211	0.360	0.330	0.088
	std	0.035	0.054	0.048	0.016	0.015	0.023	0.035	0.032	0.042	0.030	0.033	0.029
$L_{2,1}$-NMF	mean	0.474	0.439	0.213	0.493	0.462	0.234	0.482	0.433	0.210	0.375	0.328	0.091
	std	0.031	0.040	0.038	0.023	0.028	0.023	0.024	0.014	0.025	0.031	0.034	0.025
CaNMF	mean	0.518	0.473	0.255	0.522	**0.504**	0.246	0.528	0.495	0.193	0.406	0.352	0.131
	std	0.008	0.011	0.006	0.042	0.047	0.047	0.024	0.035	0.017	0.013	0.019	0.012
Hx-NMF	mean	0.495	0.471	0.238	0.477	0.447	0.210	0.484	0.462	0.220	0.374	0.347	0.093
	std	0.023	0.028	0.023	0.021	0.035	0.026	0.021	0.027	0.031	0.020	0.025	0.016
rCIM-NMF	mean	0.494	0.445	0.233	0.310	0.281	0.055	0.472	0.429	0.204	0.355	0.316	0.074
	std	0.027	0.030	0.032	0.019	0.018	0.007	0.026	0.024	0.034	0.033	0.030	0.026
Huber-NMF	mean	0.480	0.446	0.224	0.475	0.444	0.208	0.489	0.448	0.229	0.356	0.326	0.079
	std	0.026	0.015	0.027	0.020	0.021	0.028	0.016	0.015	0.016	0.024	0.016	0.017
L1-NMF	mean	0.513	0.475	0.254	0.451	0.378	0.170	0.462	0.419	0.191	0.362	0.321	0.078
	std	0.033	0.029	0.040	0.020	0.015	0.014	0.017	0.011	0.017	0.015	0.018	0.012
CauchyNMF	mean	**0.533**	0.502	**0.290**	0.485	0.479	0.210	0.508	0.457	0.251	0.385	0.339	0.099
	std	0.026	0.032	0.033	0.023	0.035	0.034	0.055	0.052	0.073	0.008	0.015	0.014
Truncated CauchyNMF	mean	0.530	0.503	0.275	0.522	0.491	0.272	0.509	0.470	0.250	0.397	0.355	0.115
	std	0.031	0.026	0.036	0.027	0.022	0.035	0.007	0.022	0.010	0.052	0.047	0.049
SPLNMF	mean	0.489	0.468	0.206	0.482	0.454	0.204	0.488	0.465	0.213	0.377	0.331	0.087
	std	0.014	0.014	0.015	0.005	0.008	0.004	0.032	0.031	0.035	0.013	0.009	0.010
ENMF	mean	0.503	0.452	0.241	0.496	0.467	0.228	0.491	0.456	0.226	0.378	0.339	0.098
	std	0.038	0.038	0.048	0.031	0.027	0.037	0.018	0.028	0.030	0.032	0.037	0.028
Hard SE-NMF	mean	0.531	**0.509**	0.282	0.523	0.502	**0.275**	**0.536**	**0.489**	0.258	**0.414**	**0.366**	**0.134**
	std	0.034	0.018	0.038	0.026	0.034	0.029	0.027	0.024	0.033	0.006	0.003	0.004
Soft SE-NMF	mean	0.512	0.479	0.256	**0.532**	0.499	0.269	0.525	0.483	**0.273**	0.407	0.349	0.118
	std	0.024	0.032	0.035	0.023	0.037	0.028	0.027	0.003	0.025	0.031	0.034	0.032

- **Hx-NMF** [27]: As a modification of NMF, Hx-NMF utilizes a sample-wise logarithmic loss function to be robust to outlier.
- **Cauchy NMF** [28]: Element-wise NMF model which considers an isotropic Cauchy distribution to evaluate the reconstruction error.
- **Truncated Cauchy NMF** [29]: This robust model learns the representation of the data robustly based on the Truncated Cauchy loss.
- **SPLNMF** [36]: A robust NMF method that incorporates the self-paced learning (SPL) into $L_{2,1}$ norm based NMF for avoiding a bad local minima.
- **Elastic NMF** [23]: ENMF proposes an elastic loss which is intercalated between Frobenius norm and $L_{2,1}$ norm.

E. Results

Tables III–VII show the clustering results on five image datasets in the four different cases, including clean, Gaussian, Laplacian, and contiguous occlusion noisy datasets. Moreover, to compare and verify our method, we conducted three widely metrics in term of clustering namely NMI, ACC, and ARI on the all methods. Also, to verify the effective of hard and soft weighting schemes, we perform both hard SE-NMF (12) and Soft SE-NMF (16) models. In tables III–VII, the best results are marked in bold face and the second-best results are highlighted in underline. Based on these tables, SE-NMF models ranked first in 49 out of 60 cases and second for the rest. It can be concluded that compared to other methods, the proposed method is able to cover a wide range of noise models. The proposed model not only presents better robustness on noisy datasets, but also almost outperforms all compared NMF family on the clean image datasets. The superiority of our method shows that by leveraging the power of both L_2 and L_1 minimizations with the modified self-paced learning method, our model can learn a more discriminative representation.

In addition, the proposed Soft SE-NMF is better than that of Hard SE-NMF, which verifies that the clustering performance can be further improved by utilizing the soft weighting scheme. Moreover, our model compared with elastic NMF, that considers the same basic loss functions (i.e., Frobenius norm and $L_{2,1}$ Norm), performs better in all cases. In the elastic NMF, the large initial sample errors lead to learning model by $L_{2,1}$ loss predominantly, hence, this model is unable to utilize the Frobenius loss perfectly. This drawback results poor performance compare with the parametric loss such as Cauchy loss and Truncated Cauchy loss in our experiments. However, the Self-Elastic NMF by a modified self-paced learning is able to fuse $L_{2,1}$ norm and Frobenius norm more effectively. Therefore, this adaptive loss outperforms other non-parametric and parametric losses.

F. Robustness on various noise versions

In this subsection, to evaluate robustness of SE-NMF, two types of noises including Gaussian and Laplacian noises with different ranges on Yale and ORL datasets are applied. The tasks can be challenging since it requires filtering out plenty of outliers with large magnitudes in order to extract a clean subspace. For Gaussian noise version, we simulate Gaussian noise from its distribution with mean 0 and deviation 0.3 and ranges of contaminated pixels in each image are various from 40% to 90%. For Laplacian noise, deviation parameter in its distribution are selected from $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. Figure 5 indicates some instance on the ORL and Yale datasets with both Gaussian and Laplace noise versions. From figure 6, it can be observed that, as the noise level rises, the soft SE-NMF method has, in most cases, the best clustering results compared to other methods. This result shows the proposed

TABLE IV
THE COMPARISON RESULTS ON THE ORL DATASET FOR DIFFERENT TYPES OF NOISE, EVALUATED BASED ON NMI, ACC, AND ARI METRICS.

methods		Clean			Gaussian			Laplacian			Occluded		
		NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
NMF	mean	0.791	0.666	0.467	0.790	0.663	0.466	0.787	0.659	0.456	0.448	0.231	0.039
	std	0.010	0.018	0.029	0.026	0.044	0.059	0.004	0.015	0.012	0.008	0.012	0.011
$L_{2,1}$-NMF	mean	0.799	0.682	0.492	0.802	0.679	0.497	0.792	0.678	0.473	0.461	0.240	0.058
	std	0.008	0.021	0.018	0.013	0.021	0.033	0.008	0.014	0.023	0.010	0.012	0.015
CaNMF	mean	0.822	0.718	0.534	0.799	0.681	0.484	0.774	0.654	0.447	0.476	0.253	0.070
	std	0.012	0.021	0.030	0.006	0.008	0.022	0.006	0.012	0.007	0.006	0.008	0.006
Hx-NMF	mean	0.818	0.713	0.529	0.814	0.704	0.522	0.788	0.673	0.468	0.459	0.238	0.056
	std	0.019	0.031	0.039	0.019	0.039	0.044	0.011	0.018	0.025	0.009	0.017	0.011
rCIM-NMF	mean	0.807	0.695	0.505	0.768	0.630	0.407	0.768	0.634	0.412	0.464	0.245	0.056
	std	0.015	0.025	0.028	0.016	0.023	0.025	0.007	0.016	0.010	0.014	0.016	0.015
Huber-NMF	mean	0.806	0.690	0.499	0.797	0.679	0.474	0.768	0.645	0.424	0.452	0.226	0.042
	std	0.008	0.015	0.026	0.013	0.020	0.027	0.016	0.018	0.034	0.012	0.012	0.013
L_1-NMF	mean	0.767	0.632	0.433	0.736	0.593	0.383	0.719	0.571	0.351	**0.522**	**0.312**	**0.101**
	std	0.020	0.022	0.028	0.017	0.024	0.030	0.006	0.010	0.015	0.005	0.012	0.004
CauchyNMF	mean	0.822	0.715	0.532	0.807	0.694	0.505	0.797	0.683	0.472	0.462	0.244	0.055
	std	0.012	0.018	0.021	0.013	0.015	0.025	0.004	0.005	0.009	0.011	0.020	0.014
Truncated CauchyNMF	mean	0.821	0.713	0.536	0.812	0.703	0.512	0.796	0.672	0.479	0.468	0.245	0.061
	std	0.014	0.025	0.027	0.014	0.010	0.031	0.005	0.012	0.017	0.005	0.009	0.007
SPLNMF	mean	0.731	0.606	0.363	0.711	0.576	0.321	0.741	0.607	0.383	0.471	0.256	0.062
	std	0.011	0.021	0.020	0.017	0.021	0.026	0.031	0.033	0.049	0.003	0.004	0.003
ENMF	mean	0.808	0.694	0.507	0.803	0.691	0.490	0.787	0.673	0.472	0.459	0.232	0.050
	std	0.017	0.033	0.041	0.015	0.025	0.035	0.018	0.021	0.036	0.012	0.014	0.010
Hard SE-NMF	mean	**0.832**	**0.738**	**0.564**	**0.823**	0.706	**0.535**	0.797	0.684	0.494	0.473	0.256	0.074
	std	0.021	0.035	0.043	0.004	0.014	0.015	0.016	0.031	0.040	0.016	0.017	0.020
Soft SE-NMF	mean	0.827	0.728	0.547	0.818	**0.721**	0.530	**0.808**	**0.701**	**0.512**	0.476	0.252	0.072
	std	0.007	0.012	0.010	0.025	0.036	0.045	0.028	0.039	0.057	0.015	0.023	0.018

TABLE V
THE COMPARISON RESULTS ON THE COIL20 DATASET FOR DIFFERENT TYPES OF NOISE, EVALUATED BASED ON NMI, ACC, AND ARI METRICS.

methods		Clean			Gaussian			Laplacian			Occluded		
		NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
NMF	mean	0.760	0.694	0.584	0.756	0.675	0.564	0.749	0.669	0.567	0.590	0.527	0.373
	std	0.016	0.026	0.030	0.021	0.031	0.030	0.010	0.012	0.010	0.035	0.042	0.049
$L_{2,1}$-NMF	mean	0.764	0.692	0.586	0.755	0.678	0.560	0.749	0.668	0.564	0.596	0.547	0.391
	std	0.030	0.031	0.051	0.016	0.021	0.030	0.018	0.023	0.031	0.014	0.017	0.022
CaNMF	mean	0.775	0.711	0.610	0.779	0.714	0.605	0.753	0.689	0.573	0.607	0.555	0.398
	std	0.013	0.016	0.025	0.007	0.015	0.021	0.010	0.026	0.022	0.011	0.020	0.019
Hx-NMF	mean	0.747	0.677	0.563	0.748	0.663	0.560	0.752	0.669	0.560	0.595	0.540	0.385
	std	0.017	0.031	0.024	0.017	0.013	0.023	0.008	0.024	0.024	0.015	0.007	0.011
rCIM-NMF	mean	0.759	0.677	0.574	0.727	0.629	0.461	0.742	0.659	0.549	0.599	0.549	0.391
	std	0.018	0.034	0.031	0.031	0.033	0.072	0.018	0.024	0.029	0.010	0.007	0.008
Huber-NMF	mean	0.767	0.694	0.592	0.753	0.681	0.559	0.763	0.688	0.582	0.600	0.544	0.390
	std	0.024	0.035	0.040	0.017	0.016	0.027	0.009	0.016	0.016	0.018	0.017	0.023
L_1-NMF	mean	0.743	0.678	0.561	0.759	0.688	0.577	0.738	0.666	0.556	0.470	0.409	0.260
	std	0.016	0.025	0.029	0.013	0.019	0.022	0.022	0.037	0.039	0.068	0.063	0.079
CauchyNMF	mean	0.768	0.699	0.602	0.771	0.695	0.588	0.766	0.693	0.596	0.609	0.548	0.403
	std	0.024	0.015	0.034	0.017	0.016	0.025	0.019	0.039	0.038	0.015	0.020	0.026
Truncated CauchyNMF	mean	0.776	0.703	0.599	0.783	**0.726**	0.620	0.772	0.701	0.598	0.618	0.562	0.405
	std	0.017	0.025	0.036	0.003	0.003	0.010	0.021	0.012	0.025	0.013	0.013	0.018
SPLNMF	mean	0.689	0.609	0.463	0.695	0.625	0.452	0.715	0.653	0.526	0.437	0.365	0.210
	std	0.002	0.003	0.003	0.003	0.006	0.003	0.008	0.002	0.012	0.002	0.006	0.002
ENMF	mean	0.764	0.691	0.583	0.766	0.701	0.589	0.772	0.697	0.595	0.605	0.553	0.403
	std	0.019	0.038	0.035	0.014	0.015	0.031	0.018	0.025	0.036	0.022	0.031	0.027
Hard SE-NMF	mean	0.779	0.714	0.611	0.784	0.719	**0.623**	**0.782**	**0.707**	**0.614**	0.627	0.569	0.430
	std	0.020	0.008	0.033	0.009	0.016	0.018	0.017	0.022	0.025	0.021	0.033	0.017
Soft SE-NMF	mean	**0.781**	**0.719**	**0.625**	**0.787**	0.717	0.617	0.774	0.694	0.597	**0.628**	**0.586**	**0.434**
	std	0.006	0.002	0.010	0.016	0.027	0.008	0.020	0.020	0.019	0.010	0.009	0.006

model is more robust and insusceptible than the methods being compared in the ORL and Yale datasets with both types of noise.

G. Parameter sensitivity

Hyperparameter tuning for unsupervised learning is always a difficult task. In this subsection, the hyperparameter sensitivity is conducted on MNIST dataset with Gaussian and Laplacian noises which demonstrates influence of two hyperparameters on the performance of SE-NMF. In SE-NMF model, there are two hyperparameters: pace parameter threshold λ and growth parameter μ. The values of λ are searched in the range of $[0.5, 12.5]$ with the step size $\Delta\lambda$ = 0.5 and the values of μ are selected from $\{1.001, 1.002, 1.004, 1.006, 1.008, 1.01\}$. From figure 7 , one can see that, the SE-NMF model achieves better NMI performance when the parameter λ relatively is neither too large or too small (when it is within the range [8, 10]) and the best parameter values for μ are almost {1.001, 1.002, 1.004}. From this analysis, it can be seen that the proposed method is relatively insensitive to hyperparameters.

TABLE VI
THE COMPARISON RESULTS ON THE MNIST DATASET FOR DIFFERENT TYPES OF NOISE, EVALUATED BASED ON NMI, ACC, AND ARI METRICS.

methods		Clean			Gaussian			Laplacian			Occluded		
		NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
NMF	mean	0.433	0.531	0.285	0.443	0.547	0.298	0.432	0.539	0.294	0.220	0.319	0.105
	std	0.028	0.032	0.034	0.025	0.031	0.026	0.023	0.031	0.028	0.023	0.019	0.017
$L_{2,1}$-NMF	mean	0.435	0.537	0.288	0.460	0.563	0.321	0.459	0.558	0.318	0.222	0.312	0.103
	std	0.002	0.032	0.014	0.010	0.015	0.015	0.008	0.025	0.025	0.017	0.020	0.014
CaNMF	mean	0.427	0.512	0.274	0.445	0.538	0.294	0.442	0.547	0.294	0.213	0.300	0.096
	std	0.022	0.023	0.024	0.020	0.027	0.021	0.011	0.029	0.021	0.014	0.014	0.009
Hx-NMF	mean	0.440	0.526	0.275	0.439	0.530	0.279	0.443	0.546	0.304	0.208	0.303	0.093
	std	0.014	0.009	0.015	0.019	0.012	0.020	0.019	0.030	0.029	0.009	0.008	0.004
rCIM-NMF	mean	0.435	0.536	0.286	0.423	0.488	0.252	0.458	0.559	0.317	0.214	0.314	0.100
	std	0.021	0.034	0.029	0.014	0.016	0.019	0.021	0.025	0.029	0.020	0.023	0.014
Huber-NMF	mean	0.442	0.539	0.296	0.432	0.532	0.285	0.440	0.546	0.302	0.213	0.309	0.099
	std	0.014	0.033	0.024	0.024	0.035	0.032	0.026	0.037	0.034	0.008	0.009	0.006
$L-1$-NMF	mean	0.381	0.485	0.240	0.387	0.493	0.253	0.392	0.494	0.249	0.180	0.280	0.075
	std	0.042	0.032	0.039	0.015	0.014	0.012	0.014	0.032	0.024	0.021	0.012	0.009
CauchyNMF	mean	0.463	0.550	0.316	0.452	0.548	0.304	0.454	0.557	0.316	0.235	0.326	0.111
	std	0.026	0.031	0.038	0.013	0.021	0.018	0.023	0.023	0.024	0.016	0.020	0.010
Truncated CauchyNMF	mean	0.466	0.557	**0.317**	0.471	0.568	0.327	**0.473**	**0.581**	**0.338**	0.236	0.338	0.118
	std	0.012	0.027	0.024	0.011	0.012	0.009	0.015	0.018	0.027	0.007	0.016	0.005
SPLNMF	mean	0.426	0.552	0.295	0.438	0.557	0.322	0.436	0.575	0.318	0.222	0.347	0.121
	std	0.006	0.002	0.003	0.011	0.008	0.014	0.001	0.001	0.006	0.004	0.012	0.008
ENMF	mean	0.438	0.533	0.293	0.457	0.552	0.308	0.452	0.553	0.315	0.228	0.320	0.107
	std	0.009	0.017	0.012	0.003	0.022	0.020	0.025	0.028	0.036	0.013	0.011	0.009
Hard SE-NMF	mean	0.472	0.569	0.314	0.475	0.570	**0.330**	0.469	0.565	0.330	0.238	0.332	0.122
	std	0.009	0.006	0.015	0.008	0.028	0.014	0.037	0.040	0.046	0.005	0.010	0.006
Soft SE-NMF	mean	**0.474**	**0.573**	0.308	**0.479**	**0.587**	0.327	0.468	0.564	0.324	**0.244**	**0.349**	**0.125**
	std	0.015	0.023	0.009	0.013	0.018	0.006	0.015	0.031	0.012	0.008	0.004	0.011

TABLE VII
THE COMPARISON RESULTS ON THE FASHION-MNIST DATASET FOR DIFFERENT TYPES OF NOISE, EVALUATED BASED ON NMI, ACC, AND ARI METRICS.

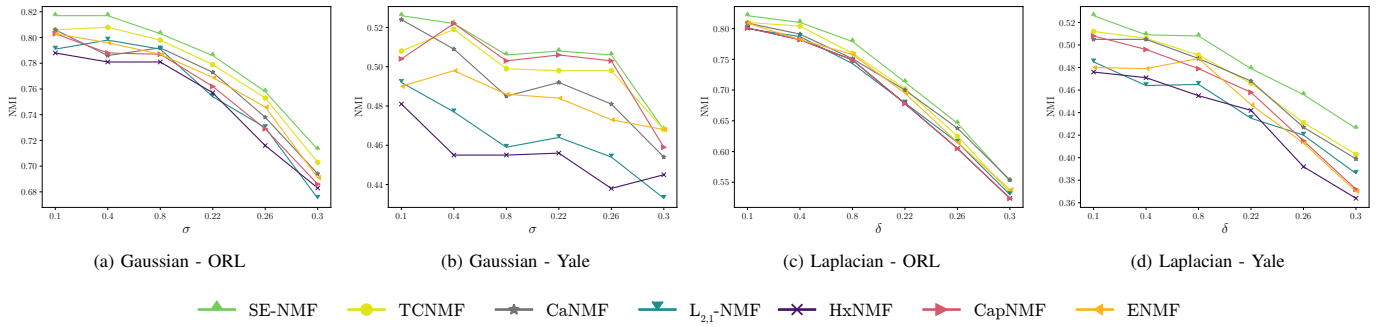
methods		Clean			Gaussian			Laplacian			Occluded		
		NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
NMF	mean	0.555	0.617	0.403	0.543	0.593	0.380	0.574	0.626	0.424	0.407	0.472	0.254
	std	0.013	0.019	0.020	0.010	0.011	0.012	0.017	0.035	0.038	0.017	0.017	0.018
$L_{2,1}$-NMF	mean	0.543	0.587	0.374	0.560	0.621	0.410	0.579	0.633	0.440	0.387	0.464	0.241
	std	0.026	0.044	0.040	0.021	0.032	0.038	0.027	0.040	0.039	0.027	0.027	0.019
CaNMF	mean	0.543	0.612	0.393	0.543	0.604	0.386	0.564	0.636	0.433	0.387	0.451	0.235
	std	0.010	0.016	0.012	0.009	0.023	0.023	0.030	0.040	0.039	0.010	0.017	0.013
Hx-NMF	mean	0.531	0.571	0.357	0.544	0.595	0.376	0.571	0.635	0.431	0.400	0.461	0.249
	std	0.031	0.045	0.053	0.012	0.042	0.028	0.014	0.017	0.021	0.015	0.014	0.011
rCIM-NMF	mean	0.544	0.586	0.383	0.550	0.581	0.361	0.579	0.613	0.423	0.388	0.450	0.233
	std	0.026	0.041	0.031	0.014	0.028	0.024	0.007	0.026	0.023	0.023	0.020	0.023
Huber-NMF	mean	0.538	0.585	0.374	0.550	0.606	0.394	0.579	0.631	0.435	0.392	0.455	0.247
	std	0.027	0.044	0.048	0.012	0.025	0.028	0.025	0.035	0.031	0.018	0.023	0.022
L_1-NMF	mean	0.510	0.570	0.353	0.542	0.611	0.391	0.569	0.638	0.433	0.254	0.331	0.130
	std	0.008	0.023	0.017	0.013	0.030	0.024	0.011	0.016	0.011	0.011	0.018	0.011
CauchyNMF	mean	0.569	0.633	0.424	0.571	0.633	0.423	0.582	0.621	0.421	0.413	0.456	0.262
	std	0.026	0.050	0.048	0.018	0.030	0.029	0.011	0.011	0.006	0.013	0.026	0.016
Truncated CauchyNMF	mean	0.578	0.640	0.432	0.571	0.629	0.420	0.591	0.640	0.442	0.426	0.476	0.274
	std	0.015	0.036	0.031	0.010	0.019	0.019	0.009	0.020	0.018	0.002	0.010	0.006
SPLNMF	mean	0.565	0.630	0.441	0.548	0.610	0.415	0.551	0.593	0.412	0.383	0.439	0.273
	std	0.004	0.005	0.008	0.009	0.011	0.007	0.002	0.003	0.002	0.001	0.004	0.003
ENMF	mean	0.562	0.631	0.417	0.549	0.602	0.391	0.578	0.619	0.421	0.411	0.472	0.262
	std	0.026	0.035	0.036	0.012	0.018	0.015	0.001	0.017	0.014	0.012	0.018	0.017
Hard SE-NMF	mean	0.583	0.645	0.444	**0.578**	**0.640**	**0.434**	0.595	**0.657**	**0.460**	0.429	0.482	0.281
	std	0.011	0.011	0.016	0.016	0.027	0.021	0.011	0.022	0.018	0.002	0.004	0.009
Soft SE-NMF	mean	**0.584**	**0.647**	**0.445**	0.575	0.636	0.429	**0.596**	0.646	0.445	**0.433**	**0.495**	**0.288**
	std	0.019	0.024	0.019	0.012	0.036	0.028	0.015	0.018	0.018	0.009	0.008	0.006

H. Analysis SE-NMF terms contribution

To analyze and show the ability of the proposed model in setting the contiribution of each loss for input samples during the learning process, we demonstrate this process by setting up a pace-by-pace experiment. We conducted SE-NMF model (with fixed parameters) on the MNIST dataset contaminated by three various deviation of Laplacian noise, and also reported NMI measure per iteration. In figure 8, x-axis indicates the iteration number, and y-axis signifies the contribution degree of L_2 and L_1 loss functions. From figure 8 (a), we can derive that since the magnitude of Laplacian noise is small, our model tends to increase the contribution of L_2 loss faster, and more samples are allowed to be learnt by L_2 loss. In figure 8 (b) and (c), due to the existence of large magnitudes of Laplacian noise, the SE-NMF model treats conservatively, and the self-paced mechanism contributes fewer samples to be learnt by L_2 loss. As a result, we can derive that our model is able to learn subspace with different magnitudes of noises in input data adaptively.

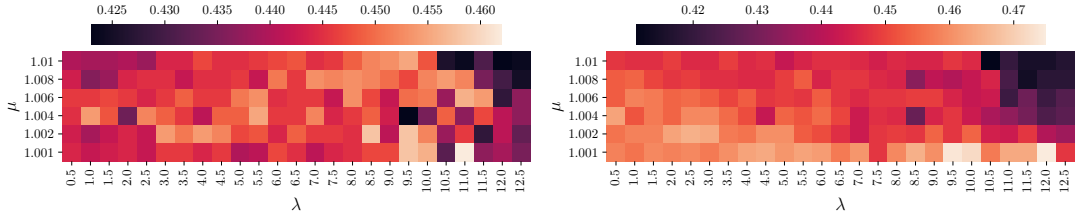
(a) Examples of the ORL dataset with Gaussian noise (b) Examples of the Yale dataset with Gaussian noise (c) Examples of the ORL dataset with Laplacian noise (d) Examples of the Yale dataset with Laplacian noise

Fig. 5. Example of image dataset with different Gaussian and Laplacian noise intensities.



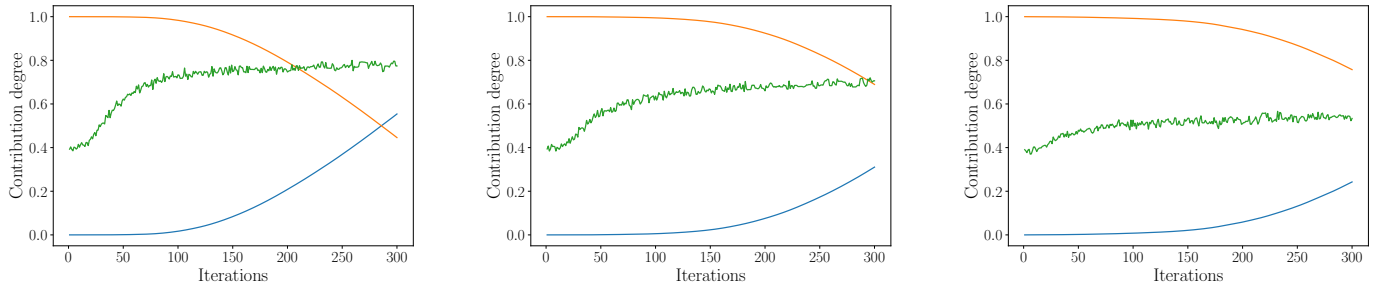
(a) Gaussian - ORL (b) Gaussian - Yale (c) Laplacian - ORL (d) Laplacian - Yale

Fig. 6. Results on the ORL and Yale datasets with different Gaussian and Laplacian noise intensities.



(a) MNIST dataset with Gaussian noise (b) MNIST dataset with Laplacian noise

Fig. 7. Parameter analysis of the SE-NMF model based on λ and μ parameters using NMI metric.



(a) Laplacian noise ($\delta = 0.1$) (b) Laplacian noise ($\delta = 0.2$) (c) Laplacian noise ($\delta = 0.3$)

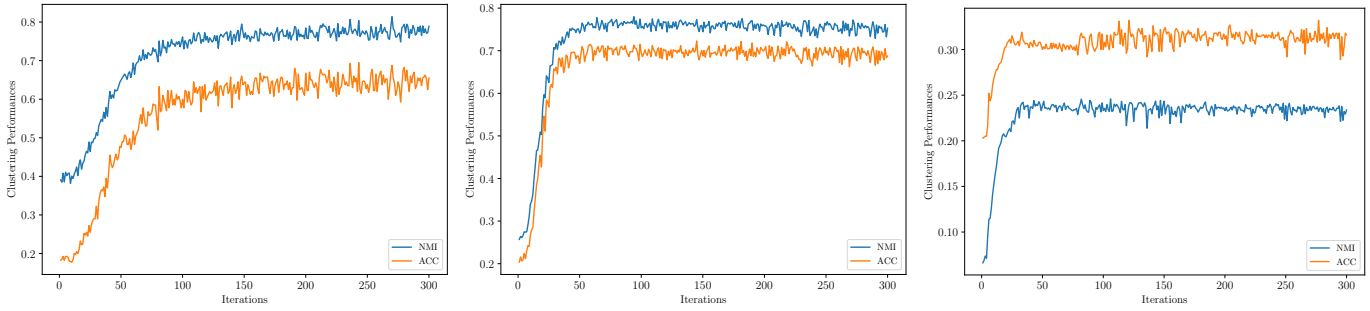
Fig. 8. The behavior of the proposed self-paced mechanism on MNIST dataset with different Laplacian noise intensities. The x-axis and y-axis indicate the iteration number and the contribution degree of loss functions, respectively. The blue and orange lines show the contribution of L_2 and L_1 losses, and green line indicates the NMI value.

I. Convergence analysis

The updating procedures for optimizing cost function of the SE-NMF are fundamentally iterative, and convergence is theoretically assured. In this section, the ACC and NMI trend curves with regard to SE-NMF iterations are displayed. Since the results from other datasets often have a similar trend, we merely provide results from the datasets Yale, COIL20, and Fashion-MNIST. The clustering results in terms of ACC and NMI measures versus the iteration numbers are illustrated in Figure 9 , where the y-axis indicate the clustering performances and the x-axis indicates the number of iterations. As shown in Figure 9 , the clustering performances in terms of ACC and NMI improve as the number of iterations rises after the initial few iterations. Additionally, the repetitive update rules for our approach converge rapidly, showing the effectiveness of the proposed updating algorithm.

V. Conclusion

This paper proposes a self-paced elastic loss function to increase the robustness of NMF model on data representation. The Self-Elastic NMF model trades-off between Frobenius norm and $L_{2,1}$ norm according to a curriculum for evaluating



(a) ORL-Laplacian (b) COIL20-Gaussian (c) MNIST-occluded

Fig. 9. Convergence analysis of the Self-Elastic NMF model on the three datasets with different noise types.

the quality of factorization, hence it is more robust to a wide-range of noises. Also, the soft weighting strategy was incorporated into self-paced learning to improve the performance of our model. An iterative updating algorithm is presented to solve the optimization problem of our model. We experimentally evaluate the robustness and efficacy of our approach on both clean and noisy datasets and we demonstrated that SE-NMF is significantly robust for learning subspace even when a portion of the samples is contaminated.

Several directions can be investigated in future work: (1) By incorporating more diverse loss functions, the SE-NMF model can be expanded into a distributionally robust learning model to enhance its robustness against a wide range of noise distributions; (2) A truncated Self-Elastic NMF can be extended which can simultaneously and appropriately model both noisy data and extreme outliers by filtering out large residual losses; (3) In addition to deep algebraic models, our loss function is usable in deep neural models where overfitting to noisy data could be an issue.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[3] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[4] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[5] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[6] P. Spurek, J. Tabor, M. Śmieja *et al.*, "Fast independent component analysis algorithm with a simple closed-form solution," *Knowledge-Based Systems*, vol. 161, pp. 26–34, 2018.

[7] W. Yan, B. Zhang, S. Ma, and Z. Yang, "A novel regularized concept factorization for document clustering," *Knowledge-Based Systems*, vol. 135, pp. 147–158, 2017.

[8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[9] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 535–541, 2000.

[10] S. A. Seyedi, F. Akhlaghian Tab, A. Lotfi, N. Salahian, and J. Chavoshinejad, "Elastic adversarial deep nonnegative matrix factorization for matrix completion," *Information Sciences*, vol. 621, pp. 562–579, 2023.

[11] Z. Shajarian, S. A. Seyedi, and P. Moradi, "A clustering-based matrix factorization method to improve the accuracy of recommendation systems," in *2017 Iranian Conference on Electrical Engineering (ICEE)*, 2017, pp. 2241–2246.

[12] J. Chavoshinejad, S. A. Seyedi, F. Akhlaghian Tab, and N. Salahian, "Self-supervised semi-supervised nonnegative matrix factorization for data clustering," *Pattern Recognition*, vol. 137, p. 109282, 2023.

[13] N. Salahian, F. A. Tab, S. A. Seyedi, and J. Chavoshinejad, "Deep autoencoder-like nmf with contrastive regularization and feature relationship preservation," *Expert Systems with Applications*, vol. 214, p. 119051, 2023.

[14] S. A. Seyedi, P. Moradi, and F. A. Tab, "A weakly-supervised factorization method with dynamic graph embedding," in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, pp. 213–218.

[15] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2014.

[16] R. Abdollahi, S. Amjad Seyedi, and M. Reza Noorimehr, "Asymmetric semi-nonnegative matrix factorization for directed graph clustering," in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2020, pp. 323–328.

[17] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 606–610.

[18] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 126–135.

[19] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[20] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[21] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1299–1311, 2012.

[22] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 673–682.

[23] H. Xiong and D. Kong, "Elastic nonnegative matrix factorization," *Pattern Recognition*, vol. 90, pp. 464–475, 2019.

[24] Q. Ke and T. Kanade, "Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *IEEE Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 739–746.

[25] H. Gao, F. Nie, W. Cai, and H. Huang, "Robust capped norm nonnegative matrix factorization: Capped norm nmf," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 871–880.

[26] L. Du, X. Li, and Y.-D. Shen, "Robust nonnegative matrix factorization via half-quadratic minimization," in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 201–210.

[27] Q. Wang, X. He, X. Jiang, and X. Li, "Robust bi-stochastic graph regularized matrix factorization for data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 390–403, 2022.
[28] A. Liutkus, D. Fitzgerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5.
[29] N. Guan, T. Liu, Y. Zhang, D. Tao, and L. S. Davis, "Truncated cauchy non-negative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 246–259, 2019.
[30] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," vol. 23, pp. 1189–1197, 2010.
[31] H. Li and M. Gong, "Self-paced convolutional neural networks." in *IJCAI*, 2017, pp. 2110–2116.
[32] J. S. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2379–2386.
[33] Z. Huang, Y. Ren, X. Pu, L. Pan, D. Yao, and G. Yu, "Dual self-paced multi-view clustering," *Neural Networks*, vol. 140, pp. 184–192, 2021.
[34] S. A. Seyedi, S. S. Ghodsi, F. Akhlaghian, M. Jalili, and P. Moradi, "Self-paced multi-label learning with diversity," in *Proceedings of The Eleventh Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 101, 2019, pp. 790–805.
[35] X. Zhu and Z. Zhang, "Improved self-paced learning framework for nonnegative matrix factorization," *Pattern Recognition Letters*, vol. 97, pp. 1–7, 2017.
[36] S. Huang, P. Zhao, Y. Ren, T. Li, and Z. Xu, "Self-paced and soft-weighted nonnegative matrix factorization for data representation," *Knowledge-Based Systems*, vol. 164, pp. 29–37, 2019.
[37] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014, pp. 2078–2086.
[38] X. Lin and P. C. Boutros, "Optimization and expansion of non-negative matrix factorization," *BMC Bioinformatics*, vol. 21, no. 1, pp. 1–10, Dec. 2020.
[39] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
[40] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear algebra and its applications*, vol. 416, no. 1, pp. 29–47, 2006.
[41] A. Robey, L. Chamon, G. J. Pappas, and H. Hassani, "Probabilistically robust learning: Balancing average and worst-case performance," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 667–18 686.
[42] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, pp. 494–501.
[43] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 547–556.
[44] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2000, p. 535541.
[45] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[47] L. Hubert and P. Arabie, "Comparing partitions journal of classification 2 193–218," *Google Scholar*, pp. 193–128, 1985.
[48] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
[49] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.
[50] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.
[51] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.
[52] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[53] Y. Jia and T. Darrell, "Heavy-tailed distances for gradient based image descriptors," in *Neural Information Processing Systems*, vol. 24, 2011, pp. 397–405.