Pattern Recognition 169 (2026) 111732

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Instance-wise distributionally robust nonnegative matrix factorization

Wafa Barkhoda [a,b], Amjad Seyedi [c], Nicolas Gillis [c], Fardin Akhlaghian Tab [a,*]

[a] Department of Computer Engineering, University of Kurdistan, Iran
[b] Faculty of Information Technology, Kermanshah University of Technology, Iran
[c] Department of Mathematics and Operational Research, University of Mons, Belgium

ARTICLE INFO

Keywords:
Distributionally robust optimization
Nonnegative matrix factorization
Instance-wise representation
Noise distribution

ABSTRACT

Nonnegative matrix factorization (NMF) stands as a prevalent algebraic representation technique deployed across diverse domains such as data mining and machine learning. At its core, NMF aims to minimize the distance between the original input and a lower-rank approximation of it. However, when data is noisy or contains outliers, NMF struggles to provide accurate results. Existing robust methods rely on known distribution assumptions, which limit their effectiveness in real-world situations where the noise distribution is unknown. To address this gap, we introduce a new model, called instance-wise distributionally robust NMF (iDRNMF), that can handle a wide range of noise distributions. By leveraging a weighted sum multi-objective method, iDRNMF can handle multiple noise distributions and their combinations. Furthermore, while the entry-wise models assume noise contamination at the individual matrix entries level, the proposed instance-wise model assumes noise contamination at the entire data instances level (columns of the input matrix). This instance-wise model is often more appropriate for data representation tasks, as it addresses the noise affecting entire feature vectors rather than individual features. To train this model, we develop a unified multi-objective optimization framework based on an iterative reweighted algorithm, which maintains computational efficiency similar to single-objective NMFs. This framework provides flexible updating rules, making it suitable for optimizing a wide range of robust and distributionally robust objectives. Extensive experiments on various datasets with distinct noise distributions and mixtures thereof show the superior performance of iDRNMF compared to state-of-the-art models, showcasing its effectiveness in handling diverse noise profiles on real-world problems.

1. Introduction

Data representation is a fundamental technique for data analysis and machine learning and plays a crucial role in various practical applications such as computer vision, data compression, clustering, and information retrieval [1]. In these applications, the input data usually possesses a high dimensionality, making it practically challenging to learn directly from the original data. Moreover, not all features are equally important or discriminative, as many are correlated, redundant, or noisy. This leads to the necessity of obtaining a suitable representation for data with decreased dimensions. Dimensionality reduction stands out as a widely employed strategy to discover meaningful representations of data by exploring the underlying structures of data [2]. Over the past decades, matrix factorization techniques have been effectively applied to obtain such data representations. The representative matrix factorization approaches include the singular value decomposition (SVD), principal component analysis (PCA), independent component analysis (ICA), vector quantization (VQ), and nonnegative matrix factorization (NMF) [3]. Specifically, NMF has gained significant popularity for image and text representation.

NMF stands out as a classical approach for the analysis of high-dimensional nonnegative data, offering an intuitively physical interpretation [4]. Different from PCA and SVD, NMF identifies two nonnegative matrices whose product closely approximates the original matrix. Due to its restriction to additive, non-subtractive combinations, NMF acquires a natural parts-based representation as opposed to a global one for the data [3]. Typically, these two matrices consist of a basis matrix, which can be regarded as parts-based representations of the input data, and a coefficient matrix responsible for storing a low-dimensional representation. The low-dimensional representation with nonnegativity provides interpretability because the learned representation is more consistent with human perception, supported by psychological and physiological evidence [5]. Therefore, NMF has been used as a dimensionality reduction method for various applications, including clustering [6,7], community detection [8], link prediction [9],

* Corresponding author.
E-mail addresses: barkhoda@kut.ac.ir (W. Barkhoda), seyedamjad.seyedi@umons.ac.be (A. Seyedi), nicolas.gillis@umons.ac.be (N. Gillis), f.akhlaghian@uok.ac.ir (F. Akhlaghian Tab).

https://doi.org/10.1016/j.patcog.2025.111732
Received 9 September 2024; Received in revised form 27 February 2025; Accepted 16 April 2025
Available online 29 May 2025

data representation [10,11], and multi-view representation [12], across unsupervised, semi-supervised, and supervised learning paradigms.

While NMF and its extensions have demonstrated success across various domains, they face challenges in learning a robust low-dimensional representation when the original dataset is affected by outliers and noise. Basic NMF, using the least squares error function (a.k.a. the L_2 loss) to assess the quality of factorization, is optimal for datasets containing additive Gaussian noise [13]. However, it may not be the most suitable choice in the presence of non-Gaussian noise, such as heavy-tailed distributions (e.g., Laplace, Cauchy) or outliers. When dealing with non-Gaussian noise, the L_2 loss can be sensitive to extreme values and may not accurately capture the underlying patterns in the data. In such scenarios, alternative loss functions may be more effective. To address this challenge, several robust M-estimator-based loss functions have been integrated into NMF as the similarity measure, replacing the squared Euclidean distance to enhance robustness to non-Gaussian noise. Some common alternatives to the L_2 loss function are the L_1-norm [14], $L_{2,1}$-norm [13], Huber [15], and correntropy [16].

Conceptually, these methods aim to substitute the L_2 loss with a robust estimator that is less sensitive to noise and outliers. Hamza and Brady [17] introduced hypersurface cost-based NMF (HCNMF), seeking to minimize the hypersurface cost function between the data matrix and its approximation. While HCNMF represents an early effort to enhance the robustness of NMF, its optimization algorithm is time-consuming due to the complex Armijo's rule-based line search it employs. Kong et al. [13] proposed a robust $L_{2,1}$-NMF, replacing the Frobenius norm with the $L_{2,1}$ norm to measure reconstructed errors. In contrast to basic NMF, $L_{2,1}$-NMF eliminates squaring, preventing a few outliers from dominating the loss function. Moreover, $L_{2,1}$-NMF proves advantageous for data with Laplacian noise. Although this method is less sensitive to noise than HCNMF, the proposed algorithm leads to a more complicated factorization process due to its non-smooth loss function. More recently, Li et al. [18] introduced a robust NMF model based on the $L_{2,1/2}$-norm, enhancing $L_{2,1}$-NMF to improve resilience against noise and outliers.

While L_1-norm and $L_{2,1}$-norm based NMF methods show increased robustness by eliminating the square of reconstructed errors compared to the original NMF, they may still face limitations, especially in the presence of substantial outliers. Huber's function-based NMF methods also face this issue, as they involve a convolution of L_2 squared and L_1 absolute loss functions. In response to this challenge, Liutkus et al. [19] explored Cauchy NMF, which employs an isotropic Cauchy distribution to assess the reconstruction error, optimizing this process in a maximum likelihood sense. Furthermore, several robust versions of NMF have been proposed in [15], including NMF based on the Correntropy Induced Metric (CIM-NMF), row-based CIM-NMF (rCIM-NMF), and Huber-NMF. Correntropy, which is closely associated with the Welsch M-estimator. It directly corresponds to the likelihood of how similar two random variables are in the vicinity of the joint space [20].

Gao et al. [21] introduced a capped norm NMF to mitigate the impact of outliers through an outlier threshold. However, the challenge lies in the uncertainty of determining the proper outlier threshold. Guan et al. [22] proposed the three-sigma rule for outlier detection and a Truncated Cauchy loss to address outliers. The Truncated Cauchy-based NMF method exhibits insensitivity against both moderate and extreme outliers, but it requires the specification of two parameters for the truncated Cauchy function. In short, the aforementioned methods attain robustness by replacing the usual squared loss function with other measures that limit the effect of outliers on the final residue. These methods collectively are categorized under robust optimization (RO).

Different from RO, Distributionally Robust Optimization (DRO), introduced by Scarf [23], addresses data uncertainty in a probabilistic manner. It seeks not only to perform well on a fixed problem instance, parameterized by a distribution, but also simultaneously across a range of problems, each determined by a distribution within an uncertainty set Ω. DRO is the process of minimizing a worst-case expected loss function over a probabilistic ambiguity set, constructed from observed samples and characterized by certain known properties of the true data-generating distribution. The approach leads to more robust solutions. DRO has emerged as an active area of research in recent years due to its probabilistic interpretation of uncertain data, tractability, when integrated with certain metrics, and notable performance, demonstrated in numerical examples.

Gillis et al. [24] present an NMF method that incorporates distributional robustness. Their proposed model addresses the challenges associated with robust NMF by employing the β-divergence family as the objective function. The authors systematically investigate the impact of various noise distributions, including Gaussian, Poisson, and Gamma, within the context of text and audio analysis. By incorporating these distributions, the model aims to augment the accuracy and robustness of NMF across various domains. The model adopts a weighted sum of the different objective functions, including the Frobenius norm, Kullback–Leibler divergence, and Itakura-Saito divergence, thereby ensuring robustness to various types of noise distributions.

Distributionally robust NMF, despite its promising potential for handling data uncertainty, remains a relatively under-explored area with significant room for further investigation. While an entry-wise formulation has been introduced in the recent literature [24], it focuses on robustness to β-divergences within the context of text and audio analysis. Entry-wise NMF models treat each entry of the input matrix independently, making them suitable for applications where each value holds a distinct meaning [25–28], such as document-term matrices in text processing or spectrogram representations in audio analysis. A typical example is L_1–NMF ($\sum_{j=1}^{m}\sum_{i=1}^{n}|X_{ji}-[WH]_{ji}|$), which minimizes the absolute sum of individual entry-wise errors, ensuring robustness by reducing sensitivity to outliers at the element level. In contrast, instance-wise NMF models provide a holistic view of the data by treating each column as a unit, which we also refer to as an instance or a sample, making them more appropriate for datasets where entire columns represent meaningful entities, such as images in computer vision, such as images in computer vision [25], genomic and clinical data in bioinformatics [26], pixels in hyperspectral analysis [27], or time-series data in financial and insurance risk analysis [28]. Unlike entry-wise models that mitigate noise at the individual value level, instance-wise models aim to capture global structures and relationships between columns, ensuring robustness against instance-level variations. This instance-wise robustness is particularly beneficial in clustering and classification applications, where maintaining consistency across instances despite noise or outliers is crucial. A key example is $L_{2,1}$–NMF ($\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|$), which minimizes the sum of L_2 norms over columns, making the model robust at the instance level by mitigating the effect of corrupted or outlier instances while preserving the overall data structure.

In this paper, we introduce the instance-wise distributionally robust nonnegative matrix factorization (iDRNMF) model, designed to handle a wide range of noise distributions. The model is presented in a multi-objective framework, highly suitable for various robust data representation tasks. It is noteworthy that the proposed model provides a unified and general framework with a complexity order closely aligned with basic NMF. Furthermore, employing the iterative reweighted least squares method, it has the capacity to cover most objective functions found in the distributionally robust literature. The contributions of this paper are outlined as follows:

- We introduce a multi-objective NMF designed to exhibit robustness across a wide range of different types of noise distributions, termed distributionally robust, closely aligning with principles of robust optimization. To be specific, we consider a weighted sum of different objective functions, a widely employed approach in multi-objective optimization. This factorization minimizes a worst-case expected loss function over probabilistic ambiguity.

Our primary motivation for exploring this class of models stems from the inherent ambiguity in selecting a specific objective function, particularly in applications where the statistics of the noise are unknown.

- Our model focuses on the noise associated with a single sample, distinguishing it from the majority of robust NMF models that consider that the noise of each entry follows a specific distribution. Consequently, our objective function stands apart from existing models. Specifically, they evaluate the quality of factorization through an entry-wise loss function, whereas our iDRNMF model assesses this quality using an instance-wise (that is, column-wise) loss function.
- The factorization method introduced in this paper is a general framework that can handle any set of noise; however, our focus is on addressing various common distributions that are typically encountered in real-world problems, such as image data representation. We specifically consider Gaussian, Laplacian, and Cauchy loss functions, forming a comprehensive package to be robust against mesokurtic and leptokurtic noise distributions, that is, distributions without and with outliers and without and with heavy tails, respectively. To be specific, in the Laplace and the Cauchy distributions, because of their heavy tail characteristics, the data reveals a much higher degree of outliers [29]. Therefore, our iDRNMF model is adapted at handling both moderate and extreme outliers stemming from these diverse noise distributions.
- We present a unified and flexible framework to optimize any distributionally robust NMF model with a customizable set of objective functions. Our iDRNMF provides highly efficient and elegant updating rules with the use of an iterative reweighted algorithm. Notably, our model incurs nearly the same computation cost as basic NMF, making it convenient for solving problems across various contexts. Furthermore, we present an efficient and straightforward trick to enhance computational efficiency that can be easily adapted for various robust NMF models.
- To assess the distributional robustness of the proposed model, we performed a series of experiments on several well-known datasets. We employ two main evaluation settings: adding noise with distinct distributions to the data separately, and introducing a mixture of noise with different distributions, thereby simulating real-world scenarios. The experimental results show the superior performance of iDRNMF compared to state-of-the-art models.

This paper is structured as follows: Section 2 provides essential background information, offering a brief overview of standard NMF, $L_{2,1}$–NMF, and Cauchy NMF, along with an introduction to the iterative reweighted algorithm. Section 3 introduces the iDRNMF model, presenting the corresponding updating rules for the proposed model. In Section 4, the proposed method is validated on well-known real-world benchmarks.

Notation

Lowercase letters stand in for scalars, while boldface lowercase letters and uppercase letters stand in for vectors and matrices, respectively. For any matrix $\boldsymbol{M}$, its ith column and jth row are denoted by $\boldsymbol{m}_i$ and $\boldsymbol{m}^{(j)}$, and M_{ij} represents its (i, j)-entry. 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 Frobenius norm of a matrix $\boldsymbol{M} \in \mathbb{R}^{d\times n}$ is $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} M_{ji}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{M}^\top\boldsymbol{M})} = \sqrt{\mathrm{Tr}(\boldsymbol{M}\boldsymbol{M}^\top)}$. When running an algorithm to optimize a variable $\boldsymbol{x}$, the tth iterate is denoted $\boldsymbol{x}^{[t]}$.

2. Preliminaries

In this section, we provide a review of key preliminaries, including the original NMF, $L_{2,1}$-NMF, and Cauchy NMF. Additionally, we introduce the iterative reweighted algorithm, commonly employed for solving the general reconstruction problem.

2.1. Basic nonnegative matrix factorization

Let $\boldsymbol{X}$ be the given data matrix, represented as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$, where m is the number of features and n is the number of samples. Each column vector $\boldsymbol{x}_i$ denotes a nonnegative data sample with m dimensions. NMF aims to discover two nonnegative matrices $\boldsymbol{W} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{H} \in \mathbb{R}^{r\times n}$, which can accurately reconstruct the data matrix as $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$, according to the following objective function:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \sum_{i=1}^{n} \Xi\left(e(\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{h}_i)\right), \ s.t.\ \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{1}$$

where e measures the error between $\boldsymbol{x}_i$ and $\boldsymbol{W}\boldsymbol{h}_i$ and Ξ is a loss function. Each sample $\boldsymbol{x}_i$ is approximated by a linear combination of the vector bases in $\boldsymbol{W}$, using coefficients given by the vector $\boldsymbol{h}_i$. The basic NMF model utilizes the square error distance to quantify the difference between $\boldsymbol{X}$ and $\boldsymbol{W}\boldsymbol{H}$:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2, \ s.t.\ \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{2}$$

where $\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|$ is the reconstruction error of the ith sample. Although model (2) is convex in $\boldsymbol{W}$ and $\boldsymbol{H}$ separately, it loses convexity when both variables are considered simultaneously. Therefore, it is impractical to guarantee, in general, the computation of a globally optimal solution because of the NP-hardness of NMF [30], and local optimal solutions can be obtained by using standard optimization methods. It is worth noting that the majority of NMF algorithms are iterative and leverage the fact that NMF can be simplified to a convex nonnegative least squares problem (NNLS) when either $\boldsymbol{W}$ or $\boldsymbol{H}$ is fixed. Specifically, throughout each iteration, one of the two factors is held constant while the other is updated in such a way that decreases the objective function. Relies on a majorization-minimization approach, the most well-known and widely used algorithm is the Multiplicative Update Rule (MUR) [31]:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}\boldsymbol{H}^\top}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top}, \quad \boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^\top\boldsymbol{X}}{\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H}}, \tag{3}$$

where $\odot$ indicates the Hadamard product.

2.2. Iterative reweighted algorithm

The iterative reweighted least squares (IRLS) algorithm is a straightforward yet potent method commonly employed to optimize robust models [13,15,22,32,33]. Instead of directly minimizing the computationally intensive non-quadratic objective function, the IRLS method breaks it down into a sum of weighted least squares subproblems. This method leverages efficient linear algebra routines, sparse matrix operations, and parallelization for enhanced computational efficiency. Denote e_i as the reconstruction error of the ith instance, which is the difference between the true value and the reconstructed value. For example, in the context of instance-wise factorization, $e_i = \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|$ is some norm of the difference between the original instance $\boldsymbol{x}_i$ and its reconstructed version $\boldsymbol{W}\boldsymbol{h}_i$. The general reconstruction problems can be expressed as follows:

$$\min_{\boldsymbol{v}} \sum_{i=1}^{n} \Xi\left(e_i(\boldsymbol{v})\right), \tag{4}$$

where $\Xi(.)$ is an increasing function in the nonnegative scalar $e_i(v) \geq 0$, and $\boldsymbol{v} = [v_1, v_2, \dots, v_p]^\top$ contains the p unknown variables, to be computed when solving problem (4). In the factorization, $\boldsymbol{v}$ contains the entries of $\boldsymbol{W}$ and $\boldsymbol{H}$. To reach the optimal solution, it is necessary to set the derivative of problem (4) equal to zero, which is:

$$\sum_{i=1}^{n} \omega(e_i(\boldsymbol{v})) \frac{\partial e_i(\boldsymbol{v})}{\partial v_j} = 0, \quad \text{for } j = 1, 2, \dots, p, \tag{5}$$

where $\omega\left(e_i(\boldsymbol{v})\right) = \frac{d\Xi(e_i(\boldsymbol{v}))}{de_i(\boldsymbol{v})}$ is called influence function. Furthermore, Eq. (5) can be rewritten as:

$$\sum_{i=1}^{n} \psi(e_i(\boldsymbol{v}))e_i(\boldsymbol{v})\frac{\partial e_i(\boldsymbol{v})}{\partial v_j} = 0,\ j = 1, 2, \ldots, p, \tag{6}$$

where $\psi\left(e_i(\boldsymbol{v})\right)$ is called the weight function and is defined as

$$\psi\left(e(\boldsymbol{v})\right) = \frac{\omega(e(\boldsymbol{v}))}{e(\boldsymbol{v})} = \frac{\Xi'(e(\boldsymbol{v}))}{e(\boldsymbol{v})}.$$

Three main conditions ensure that the objective function has a unique solution and its optimization is computationally efficient. First, it should have a bounded influence function. Second, to ensure a unique minimum, the objective function being minimized should be convex in the parameter vector. However, even for non-convex problems like NMF, IRLS can be used to find a local minimum. Third, the objective function should have a non-zero gradient, avoiding the need to search the entire parameter space [34]. Under these conditions, the objective (6) is the solution of the following iterative reweighted problem:

$$\min_{\boldsymbol{v}} \sum_{i=1}^{n} \psi(e_i(\boldsymbol{v})^{[t-1]})e_i(\boldsymbol{v})^2, \tag{7}$$

where $e_i(\boldsymbol{v})^{[t-1]}$ represents the reconstruction error of the $(t-1)$th iteration. The process of solving problem (7) can be broken down into two steps for each iteration. Firstly, treat the weight $\psi(e_i(\boldsymbol{v})^{[t-1]})$ as a fixed value, and then choose the optimal solution based on the specific form of the problem (7). Secondly, recalculate the weight value of $\psi(e_i(\boldsymbol{v})^{[t-1]})$ based on the current reconstructed error $e_i(\boldsymbol{v})^{[t]}$. A connection was established between the iterative reweighted algorithm and NMF, regardless of the loss function used, in [32]. More precisely, the NMF loss function and $\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|$ can be considered as $\Xi(.)$ function and e_i in Eq. (4), respectively. Thus, Eq. (1) can be rewritten as:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \sum_{i=1}^{n} d_i\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 = \mathrm{Tr}[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})\,\boldsymbol{D}\,(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})^\top],$$
$$\text{s.t.}\quad \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{8}$$

where $d_i = \psi(\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^{[t-1]})$ is a coefficient computed in each iteration according to Eq. (6) and using the residue value in the previous iteration. Consequently, we can determine a diagonal matrix $\boldsymbol{D}$, which $D_{ii} = d_i$ and rewrite the update rules (3) in a weighted form as follows:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}\boldsymbol{D}\boldsymbol{H}^\top}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top}, \quad \boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^\top\boldsymbol{X}\boldsymbol{D}}{\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H}\boldsymbol{D}}. \tag{9}$$

2.3. $L_{2,1}$ nonnegative matrix factorization

The basic NMF model is sensitive to noise and outliers because it relies on squared reconstruction errors in its loss function. Large noise or outliers can significantly influence these errors, biasing the decomposition towards explaining them [32]. To improve the robustness of NMF, Kong et al. [13] introduced a more robust $L_{2,1}$-NMF, wherein the Frobenius norm was substituted with the $L_{2,1}$ norm to quantify the reconstructed errors. The objective function of $L_{2,1}$-NMF can be defined as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_{2,1} = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|, \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0,$$

where $\|.\|_{2,1}$ denotes the $L_{2,1}$-norm. The $L_{2,1}$-norm of the given matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ is defined as $\|\boldsymbol{M}\|_{2,1} = \sum_i^n \sqrt{\sum_j^m M_{ji}^2}$. Observe that $L_{2,1}$-NMF uses the original reconstructed error $\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|$, and removes the square. Consequently, $L_{2,1}$-NMF is capable of handling noise and outliers more effectively than the standard NMF. In addition, Kong et al. developed an effective iterative updating algorithm for solving $L_{2,1}$-NMF using Eq. (9) with $\boldsymbol{D}$ being:

$$D_{ii} = \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^{[t-1]}}.$$

2.4. Cauchy nonnegative matrix factorization

In standard NMF, a Gaussian distribution is assumed, whereas in $L_{2,1}$-NMF, a Laplacian distribution is considered to model the noise. Xiong et al. [33] propose to use a Cauchy distribution to model noise, represented as:

$$p\left(\boldsymbol{x}_i \,|\boldsymbol{W}\boldsymbol{h}_i\right) \sim \frac{1}{\gamma}\frac{1}{1 + \left(\frac{\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|}{\gamma}\right)^2}, \tag{10}$$

where γ is the scale parameter that determines the half-width at half-maximum (HWHM). By examining Eq. (10), it is evident that when there is a substantial difference between the observed values and the correct value (i.e., $\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|$ is large), the likelihood of noise will decrease significantly (due to its presence in the denominator). Consequently, the model tends to be more resistant to noise. Using the idea of maximizing the log-likelihood (MLE), it is possible to fit the observations correctly into the noise model given by Eq. (10). Incorporating non-negative constraints of $\boldsymbol{W} \geq 0$ and $\boldsymbol{H} \geq 0$ leads to the formulation of the Cauchy-NMF model:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_{2,cau} = \sum_{i=1}^{n} \ln(\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 + \gamma^2) \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0,$$

which can be solved via the iterative reweighted algorithm. The weighted iterative updating rules can be derived from Eqs. (9), where $\boldsymbol{D}$ is:

$$D_{ii} = \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 + \gamma^2}.$$

3. Proposed model: iDRNMF

Over time, several robust NMF models have been developed, mainly characterized by the chosen objective function that evaluates the quality of an approximation through a specific distance between $\boldsymbol{W}\boldsymbol{H}$ and $\boldsymbol{X}$. The selection of this objective function is usually determined by the assumed noise model or statistics on the data matrix $\boldsymbol{X}$. Typically, users either manually select the objective function in an ad hoc manner or it is automatically chosen using cross-validation, where training is performed on a part of the input data matrix and testing on the remaining entries [35]. Nevertheless, with these methodologies, an inappropriate choice of the objective function might result in an NMF solution that substantially diverges from the intended answer. Addressing this challenge involves designing an NMF model that is robust to different types of noise distributions and suitable across various applications.

3.1. Instance-wise distributionally robust NMF

This section introduces the instance-wise Distributionally Robust Nonnegative Matrix Factorization (iDRNMF) within a multi-objective framework, aiming to learn a robust data representation at the instance level. The proposed model is designed to handle a wide range of noise with different distributions, and its unified framework is capable of encompassing any objective function found in distributionally robust literature. Notably, in contrast to entry-wise models (e.g., L_1-NMF [14], Huber-NMF [16], CIM-NMF [15], and Cauchy NMF [19]), the proposed instance-wise model treats each column of the data matrix $\boldsymbol{X}$ as an independent sample, conducting calculations based on this premise. To be specific, we define general loss function $\|.\|_{2,\tau}$ which applies L_2 norm on each column of $\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}$ independently and subsequently computing the desired loss τ on the results. It is assumed that τ is associated with a probability distribution, allowing $\|.\|_{2,\tau}$ to accommodate a specific noise distribution. Given that the noise model on the data is unknown but corresponds to a distribution associated with a $\tau \in \Omega$, a dynamic weighted sum of different objective functions is considered, with the weights assigned to these objective functions being learned.

Therefore, the proposed solution is robust to different types of noise distributions, termed as distributionally robust, and can be tackled by optimizing a weighted sum of the different objective functions. In a min–max formulation, the goal is to minimize a worst-case expected loss function over a probabilistic ambiguity set Ω as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} \sum_{\tau\in\Omega} \lambda_\tau \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_{2,\tau}, \quad \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\ \|\lambda\|_1=1, \tag{11}$$

where $\lambda \in \mathbb{R}^{|\Omega|}$ is a weight vector and λ_τ is weight for loss τ. In this paper, we use $\Omega=\{1,2,cau\}$, which makes our model robust to Laplacian, Gaussian, and Cauchy distributions and a mixture of them. So, we could rewrite Eq. (11) as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} \lambda_1\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_{2,1}+\lambda_2\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_{2,2}^2+\lambda_{cau}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_{2,cau}, \tag{12}$$
$$\text{s.t.} \quad \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\ \|\lambda\|_1=1.$$

where $\lambda=\{\lambda_1,\lambda_2,\lambda_{cau}\}$. We can express the objective function (12) in an instance-wise representation as follows:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} & \lambda_1\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|+\lambda_2\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \\ & +\lambda_{cau}\sum_{i=1}^{n}\ln(\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2+\gamma^2), \\ \text{s.t.} \quad & \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\ \|\lambda\|_1=1.\end{aligned} \tag{13}$$

To optimize this multi-objective formulation, it is imperative to reduce the total loss in each iteration, with particular attention given to minimizing the objective with the highest loss. This approach effectively addresses worst-case scenarios. Due to the presence of a square in the Frobenius norm, it is evident that the $\|.\|_{2,2}$ norm yields significantly higher residue and $\|.\|_{2,cau}$ consistently lower ones in all iterations. This characteristic makes it hard to transition smoothly between objective functions throughout the iterations until convergence is achieved. Therefore, it is essential to scale the objectives to establish a meaningful linear combination, ensuring that each term in the sum carries equal importance. This adjustment is particularly critical for our iDRNMF model as it facilitates the generation of solutions with minimal error across all objectives, rather than prioritizing individual ones. Consequently, these solutions inherit superior qualities from those generated by various single-objective models. Similar to the DRNMF [24] model, we use the following procedure to scale different objective functions.

Initially, the problems $\zeta_\tau=\min_{\boldsymbol{W},\boldsymbol{H}\geq 0}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_{2,\tau}$ for $\tau\in\{1,2,cau\}$ are solved. Subsequently, each objective will be normalized using the corresponding error ζ_τ. We then replace (13) by substituting the objectives with their normalized forms:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} & \lambda_1\epsilon_1\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|+\lambda_2\epsilon_2\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \\ & +\lambda_{cau}\epsilon_{cau}\sum_{i=1}^{n}\ln(\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2+\gamma^2), \\ \text{s.t.} \quad & \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\ \|\lambda\|_1=1,\end{aligned} \tag{14}$$

where, for simplicity, $\epsilon_\tau=\frac{1}{\zeta_\tau}$. Notice that we use normalized objective functions because otherwise, in most cases, the above problem amounts to minimizing a specific objective corresponding to the one with the largest value. Due to this design, the proposed NMF model (14) balances the importance between all objective functions. This accomplishment stems from its ability to accurately model the ambiguity set's distribution and maintain robustness to varying noise distributions. Consequently, this model can provide a more reliable and generalizable representation across diverse scenarios. However, due to the inherent difficulty of optimizing the non-convex and non-linear objective (14), we reformulate the model as an iteratively reweighted problem.

3.2. Iterative reweighted iDRNMF

In Section 2.2, we presented an iterative reweighted algorithm and proved that based on it, we can convert any objective function into a weighted basic NMF. Therefore, by utilizing the reweighted framework (8), our multi-objective formulation (14) can also be converted as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} \sum_{i=1}^{n} d_i^{(\Omega)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2, \quad \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\|\lambda\|_1=1 \tag{15}$$

where $d_i^{(\Omega)}$ is an instance-weight and calculated according to (7) as follows:

$$d_i^{(\Omega)}=\sum_{\tau\in\Omega}\lambda_\tau\epsilon_\tau\psi_\tau(\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^{[t-1]})=\sum_{\tau\in\Omega}\lambda_\tau\epsilon_\tau d_i^{(\tau)}$$

Indeed, in problem (15) each term of the objective function is assigned a $d_i^{(\tau)}$ weight, consequently, we can rewrite it as:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} \sum_{\tau\in\Omega}\lambda_\tau\epsilon_\tau\sum_{i=1}^{n} d_i^{(\tau)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2, \quad \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\|\lambda\|_1=1,$$

where the sample weight is $d_i^{(\tau)}=\psi_\tau(\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^{[t-1]})$. If $\Omega=\{1,2,cau\}$, we have:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} & \lambda_1\epsilon_1\sum_{i=1}^{n} d_i^{(1)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2+\lambda_2\epsilon_2\sum_{i=1}^{n} d_i^{(2)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \\ & +\lambda_{cau}\epsilon_{cau}\sum_{i=1}^{n} d_i^{(cau)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\|\lambda\|_1=1,\end{aligned}$$

where $d_i^{(1)}=\frac{1}{\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|}$, $d_i^{(2)}=1$, and $d_i^{(cau)}=\frac{1}{\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2+\gamma^2}$. Finally, we can rearrange the formula as follows:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}} \max_{\lambda} & \sum_{i=1}^{n}(\lambda_1\epsilon_1 d_i^{(1)}+\lambda_2\epsilon_2+\lambda_{cau}\epsilon_{cau}d_i^{(cau)})\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \\ & =\sum_{i=1}^{n} d_i^{(\Omega)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2 \\ \text{s.t.} \quad & \boldsymbol{W},\boldsymbol{H},\lambda\geq 0,\|\lambda\|_1=1,\end{aligned} \tag{16}$$

where $d_i^{(\Omega)}=\lambda_1\epsilon_1 d_i^{(1)}+\lambda_2\epsilon_2+\lambda_{cau}\epsilon_{cau}d_i^{(cau)}$. It is important to mention that, we can expand Ω to encompass any preferred distribution and utilize this unified weighted formulation to manage it.

3.3. Optimization

Since the cost function of our iDRNMF model is not convex in $\boldsymbol{W}$ and $\boldsymbol{H}$ together, optimizing it may encounter challenges. As such, to achieve successful optimization of the factorization, it is possible to split this problem into two smaller sub-problems, both of which are convex in nature. To optimize, problem (16) can be addressed using alternating minimization (Algorithm 1), which enables us to iteratively update the variables until a satisfactory solution is achieved. In each iteration, we employ the Multiplicative Update method to update one variable while keeping the other fixed.

3.3.1. Updating factors

To achieve the update rules for $\boldsymbol{W}$ and $\boldsymbol{H}$ factors, the objective function (16) can be rewritten in the trace form to be solved by the Multiplicative Update Rule (MUR) method in a weighted NMF framework as follows:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H}} & \sum_{i=1}^{n} d_i^{(\Omega)}\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2=\mathrm{Tr}\left[(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H})^\top\right], \\ \text{s.t.} \quad & \boldsymbol{W},\boldsymbol{H}\geq 0\end{aligned} \tag{17}$$

where $D_{ii}=d_i^{(\Omega)}$ can be computed as

$$D_{ii}=\frac{\lambda_1\epsilon_1}{\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|}+\lambda_2\epsilon_2+\frac{\lambda_{cau}\epsilon_{cau}}{\|\boldsymbol{x}_i-\boldsymbol{W}\boldsymbol{h}_i\|^2+\gamma^2}. \tag{18}$$

Table 1
The count of arithmetic operations for each iteration in Basic NMF and iDRNMF.

Method		Updating $\boldsymbol{W}$	Updating $\boldsymbol{H}$	Updating $\boldsymbol{D}$	Overall
NMF	fladd	$mnr + (m+n)r^2$	$mnr + (m+n)r^2$	–	$O(mnr)$
	flmlt	$mnr + (m+n)r^2 + mr$	$mnr + (m+n)r^2 + nr$	–	
	fldiv	mr	nr	–	
iDRNMF	fladd	$mnr + (m+n)r^2$	$mnr + (m+n)r^2$	$mnr + 2n$	$O(mnr)$
	flmlt	$mnr + (m+n)r^2 + mn + (n+m)r$	$mnr + (m+n)r^2 + mn + 2nr$	$mnr + mn + 3n$	
	fldiv	mr	nr	$2n$	
	flsub	–	–	mn	

The objective function proposed in (17) has the same mathematical form as the weighted objective function given in (8). Because of this structural similarity, we can apply the weighted MURs derived in (9) to the proposed objective function (17) as follows:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}\boldsymbol{D}\boldsymbol{H}^\top}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^\top}, \tag{19}$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^\top\boldsymbol{X}\boldsymbol{D}}{\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H}\boldsymbol{D}}. \tag{20}$$

It is noteworthy to mention that the initialization of the $\boldsymbol{W}$ and $\boldsymbol{H}$ factors is carried out randomly.

3.3.2. Updating weights

The main idea to solve the optimization problem (11) is to minimize the worst-case in each iteration. Therefore, we should find the maximum value among the objective functions and try to minimize it in the next iteration. It means that in each iteration, although all objective functions are reduced, the largest one should be given more importance at the next iteration. For this reason, we use a Frank–Wolfe algorithm, which is known as the conditional gradient method [24,36]. Accordingly, we initialize the algorithm with $\lambda_\tau^{[0]} = \frac{1}{|\Omega|}$ for all $\tau \in \Omega$. Let define $p^* = \arg\max_{\tau\in\Omega} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|_{2,\tau}$ and $\lambda_*^{[t]}$ as the vector with a single non-zero entry equal to one at position p^* in tth iteration. We update all λ_τ in $(t+1)$th iteration based on the following equation:

$$\lambda^{[t+1]} = (1-\eta)\lambda^{[t]} + \eta\lambda_*^{[t]}, \tag{21}$$

According to this formula and the fact that $\|\lambda\|_1 = 1$, we can see that the corresponding λ to the maximum value among loss functions will be increased in the next iteration, while it should be decreased for the other loss functions. We can control the rate of changes with parameter η, where a larger value causes more emphasis to be paid to the maximum loss function, and on the contrary, smaller values cause slower changes in the amount of attention to it. Notice in all cases $\eta \in [0,1]$. However, a better choice can be that the value of η is not fixed and changes dynamically in the different iterations. To be specific, $\eta^{[t]}$ should have a larger value in the beginning and slightly decrease during subsequent iterations until it finally decays. In this way, the model initially pays more attention to the larger residue and while minimizing it, avoids the complete dominance of one cost function over the others. Additional discussions on how to choose η and its effect on model convergence will be presented in Section 4.7.

In our model, we adopt a strategy commonly employed in min–max optimization problems, particularly in game theory contexts. By performing only one update of the minimizing $\boldsymbol{W}$ and $\boldsymbol{H}$ factors before updating the maximizing weights λ, we follow a sequential updating process aimed at guiding convergence towards the saddle point. This strategy allows each player to react to the strategy of the others in a controlled manner, facilitating efficient optimization. However, it is important to acknowledge that excessive updates of one player before the other can introduce oscillations and instability, potentially deviating from the desired saddle point [37]. Hence, we opted for this balanced updating scheme to ensure stability and convergence in our optimization process. The proposed iterative algorithm for updating the variables in our iDRNMF formulation is presented in Algorithm 1. The source code implementation of the proposed method is publicly available at https://github.com/barkhoda/iDRNMF.

Algorithm 1 instance-wise Distributionally Robust NMF (iDRNMF)

Input: data matrix $\boldsymbol{X}$, latent factor r, a finite ambiguity set Ω;
Output: basis matrix $\boldsymbol{W}$ and representation matrix $\boldsymbol{H}$;

1: Initialize $\boldsymbol{W}$ and $\boldsymbol{H}$ randomly, $\lambda_\tau = \frac{1}{|\Omega|}, \forall\tau \in \Omega$;
2: **while** convergence not reached **do**
3: Update instance weight matrix $\boldsymbol{D}$ according to (18);
4: Update basis matrix $\boldsymbol{W}$ according to (19);
5: Update representation matrix $\boldsymbol{H}$ according to (20);
6: Update weights λ according to (21);
7: **end while**

3.4. Complexity analysis

In this section, we will delve into the computational complexity of the proposed iDRNMF algorithm, expressing its complexity using the big O notation. By comparing the update rules of iDRNMF (Eqs. (19) and (20)) with those of standard NMF (Eq. (3)), we observe that the additional computational cost primarily arises from computations related to $\boldsymbol{D}$. It is worth noting that $\boldsymbol{D}$ is a diagonal matrix, which effectively reduces the computational cost. For better understanding, we consider four types of arithmetic operations: floating-point addition (fladd), floating-point multiplication (flmlt), floating-point subtraction (flsub), and floating-point division (fldiv). Assume the number of data points is n, the number of features is m, the number of factors is r, and $\Omega = \{1, 2, cau\}$. Based on Eq. (18), the most frequent operation to compute D is multiplication, with the cost of calculating each entry D_{ii} equating to $mr + m + 3$. Additionally, computing D_{ii} involves $mr + 2$ additions, m subtraction, and 2 divisions. These costs apply to all n samples, as outlined in Table 1, which presents the final computation cost for updating $\boldsymbol{D}$. For a diagonal matrix $\boldsymbol{D}$, the operation $\boldsymbol{X}\boldsymbol{D}$ signifies that each entry D_{ii} must be multiplied with all entries in column $\boldsymbol{x}_i$. Therefore, for $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, this operation requires $m \times n$ multiplication. Consequently, the complexity of updating $\boldsymbol{W}$ and $\boldsymbol{H}$ in our iDRNMF formulation, based on Eqs. (19) and (20), is not significantly higher than that of the basic NMF method. Both methods have a complexity order $O(mnr)$ in each iteration. As a result, if the total number of iterations is t, the overall cost for both NMF and iDRNMF is $O(tmnr)$. Notably, the algorithm also scales well for sparse matrices, in $\mathcal{O}(\text{nnz}(\boldsymbol{X})r)$ operations, where $\text{nnz}(\boldsymbol{X})$ is the number of non-zero entries in $\boldsymbol{X}$, as for standard NMF. As illustrated in Table 1, the computational algorithm for iDRNMF is surprisingly straightforward, nearly matching the computational cost of standard NMF.

4. Experimental results

This section examines the robustness and efficiency of the proposed model by comparing it to nine cutting-edge robust models including $L_{2,1}$–NMF [13], FroNMF [31], CauchyNMF [33], HxNMF [32], rCIM-NMF [15], HuberNMF [15], Elastic NMF [38], Deep Autoencoder NMF (DANMF) [39], and DR-NMF [24] on seven well-known datasets. Furthermore, this study provides a comprehensive discussion on the

interpretation of results and the convergence analysis, in addition to a description of the experimental settings. To mitigate the impact of initial values, we execute each of the compared models multiple times with diverse initializations. Specifically, we run each algorithm five times and present the average results. Also, the multiplicative updating rules for factor matrices are executed in 300 iterations. The parameters for each algorithm were established based on the original articles where the methods were initially proposed. 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. Two frequently used evaluation criteria are used to evaluate the performance of clustering, including clustering accuracy (ACC) and normalized mutual information (NMI). In addition, we conduct tests using several noise models, such as Gaussian, Laplacian, Cauchy, and a combination of these, to assess and compare the effectiveness of the proposed data representation model. It is worth mentioning that our model is an auto-weighted method that does not require any hyperparameter tuning.

4.1. Datasets

In the experiments, we utilize seven real-world image datasets: two face image datasets (Yale and ORL), an object image dataset (COIL20), a handwriting image dataset (MNIST), a clothing image dataset (Fashion MNIST), and two medical datasets (OrganA-MNIST and Blood-MNIST), which have been extensively employed to evaluate the efficacy of matrix factorization models. Fig. 1 illustrates the selected samples of images in each dataset. We convert each image dataset into a data matrix, where the dimensions of the matrix correspond to the number of pixels and the number of samples in the dataset. The Yale face dataset consists of 15 individuals, with each person captured in 11 distinct images under different conditions, and for our tests, we resize each image to dimensions of 32×32 pixels. The ORL dataset consists of 400 images of 40 distinct individuals, with each person represented by 10 images captured at various instances with diverse lighting conditions and facial expressions, and we resize each image to dimensions of 32×32 pixels. The COIL20 object image dataset consists of 20 distinct objects captured from every perspective in a complete 360-degree rotation, and we adjust the size of each image to 32×32 pixels. The MNIST dataset originally contains 70,000 digital images of handwritten digits, and we employ a random selection process to choose 100 photos for each numerical digit, spanning from '0' to '9', creating a dataset of 1000 images. The Fashion-MNIST dataset serves as a replacement for the original MNIST dataset and consists of 60,000 training and 10,000 test images, each grayscale with dimensions of 28×28 pixels, representing a sample associated with one of 10 categories, and we generate the dataset by selecting 100 random samples from each class. OrganA-MNIST [40] is a dataset consisting of 58,830 abdominal CT images, each resized to 28×28 pixels, with a total of 11 organ classes. It is designed for medical image classification and deep learning applications in organ segmentation and identification. Blood-MNIST [40] is an RGB dataset consisting of 17,092 blood cell microscopy images, each resized to $28 \times 28 \times 3$ pixels, categorized into 8 different blood cell types. This dataset is useful for hematology research and deep learning tasks related to blood cell classification. Table 2 summarizes the main information about these datasets.

4.2. The combination of noise with different distributions

When we want to evaluate the robustness and efficacy of a distributionally robust model that covers noise distributions in the ambiguity set Ω, it is common to contaminate clean data with one of the covered noise distributions independently and assess the performance of the model. However, in the real world, noise does not obey a pure distribution and usually, it contaminates the data as a combination of different distributions. So, it is more realistic to evaluate the robustness of the model with these types of noise. Let

$$\hat{\boldsymbol{N}} = \sum_{\tau \in \Omega} \frac{\boldsymbol{N}_\tau}{\|\boldsymbol{N}_\tau\|_F}, \tag{22}$$

where $\boldsymbol{N}_\tau$ is noise constructed using the distribution corresponding to $\tau \in \Omega$. We set

$$\boldsymbol{N} = \varrho \frac{\|\boldsymbol{X}\|_F}{\|\hat{\boldsymbol{N}}\|_F} \hat{\boldsymbol{N}}, \ 0 < \varrho < 1, \tag{23}$$

where ϱ is the noise intensity parameter. Finally, $\hat{\boldsymbol{X}} = \max(0, \boldsymbol{X} + \boldsymbol{N})$ where $\boldsymbol{X}$ is clean data and $\hat{\boldsymbol{X}}$ is a low-rank matrix to which had been contaminated with noise.

Fig. 1. Selected samples of images in the tested datasets; from to bottom: Yale, ORL, COIL20, MNIST, Fashion-MNIST, OrganA-MNIST, and Blood-MNIST.

4.3. Clustering results

Tables 3–9 showcase the clustering results across seven image datasets in the four distinct noisy conditions: Gaussian, Laplacian, Cauchy, and their combinations. To explore the combined scenarios, we consider four combinations, namely Gaussian and Laplacian (G+L), Gaussian and Cauchy (G+C), Laplacian and Cauchy (L+C), and the combination of all noises (G+L+C). In each table, the top three performances are highlighted in boldface, underlined, and double-underlined, respectively. Additionally, we calculate the average ranking for all methods within each dataset. Remarkably, iDRNMF consistently achieves the top ranking across all datasets and noise cases. An intriguing observation is that iDRNMF consistently outperforms other methods, which have acceptable performance in some scenarios but struggle in others. iDRNMF shows a significantly better average ranking in all cases. For instance, in Table 7, Huber yields the best NMI for L+C noise, but its performance in other cases is less satisfactory, resulting in a considerably high average ranking. In summary, our proposed method exhibits superior performance, consistently ranking first across all datasets and noise cases. Notably, our approach shows its capability to handle a wide range of noise models, particularly addressing mixtures of various noises commonly encountered in real-world scenarios.

In distributionally robust models, the exact noise distribution is unknown, but it is assumed to belong to an ambiguity set Ω. Consequently, the model is anticipated to perform well for noise with any distribution $\tau \in \Omega$, without a guarantee of acceptable performance for other distributions. Nevertheless, due to our innovative combinational framework, we expect that our model exhibits acceptable performance even for noise with a distribution $\hat{\tau} \notin \Omega$. To test this hypothesis, we evaluate our model against additive Poisson noise with the mean σ,

Table 2
Details of real-world datasets under test.

Dataset	#instance	#feature (dimension)	#class	Application
Yale	165	1024 (32 × 32)	15	Face recognition
ORL	400	1024 (32 × 32)	40	Face recognition
COIL20	1440	1024 (32 × 32)	20	Object recognition
MNIST	1000	784 (28 × 28)	10	Handwriting recognition
Fashion-MNIST	1000	784 (28 × 28)	10	Clothing recognition
OrganA-MNIST	58,830	784 (28 × 28)	11	Abdominal CT
Blood-MNIST	17,092	2352 (28 × 28 × 3)	8	Blood cell microscopy

Table 3
The comparison results on the Yale dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G+L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.4285	0.4603	0.4533	0.4330	0.4898	0.4377	0.4396	5.00
	Fro	0.4506	0.4389	0.4520	0.4352	0.4632	0.4364	0.4516	5.42
	Cauchy	0.4136	0.4493	0.4448	0.4840	0.4992	0.4429	0.4524	4.71
	Hx	0.4288	0.4617	0.4559	0.4720	0.4682	0.4620	0.4450	3.85
	rCIM	0.4520	0.4658	0.4326	0.4208	0.4554	0.4359	0.4265	6.42
	Huber	0.4526	0.4580	0.4440	0.4281	0.4533	0.4282	0.4351	6.28
	Elastic	0.4467	0.4552	0.4481	0.4469	0.4890	0.4258	0.4331	5.85
	DANMF	0.4297	0.4055	0.3902	0.4030	0.4026	0.4017	0.4290	9.28
	DR-NMF	0.4671	0.4570	0.4252	0.4380	0.4513	0.4238	0.4156	7.14
	iDRNMF	**0.4914**	**0.4992**	**0.5049**	**0.4954**	**0.5131**	**0.4739**	**0.4912**	**1.00**
ACC	$L_{2,1}$	0.3806	0.4182	0.4061	0.3903	0.4364	0.3964	0.3879	5.00
	Fro	0.4024	0.4024	0.4024	0.3867	0.4158	0.3903	0.4036	5.57
	Cauchy	0.3697	0.3988	0.3952	0.4364	0.4703	0.4048	0.4125	5.00
	Hx	0.3782	0.4121	0.4145	0.4255	0.4339	0.4133	0.3988	4.42
	rCIM	0.4048	0.4327	0.3964	0.3782	0.4133	0.3842	0.3782	6.14
	Huber	0.4121	0.4097	0.3988	0.3794	0.4000	0.3806	0.3915	6.28
	Elastic	0.4061	0.4242	0.3939	0.3891	0.4436	0.3818	0.3830	5.42
	DANMF	0.4000	0.3636	0.3515	0.3636	0.3575	0.3636	0.3757	9.42
	DR-NMF	0.4085	0.4206	0.3842	0.3915	0.4012	0.3758	0.3709	6.71
	iDRNMF	**0.4230**	**0.4509**	**0.4616**	**0.4388**	**0.4727**	**0.4291**	**0.4376**	**1.00**

Table 4
The comparison results on the ORL dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G+L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.7148	0.7321	0.7322	0.7406	0.7257	0.7208	0.7157	5.57
	Fro	**0.7193**	0.7215	0.7371	0.7343	0.7210	0.7190	0.7213	6.00
	Cauchy	0.7178	0.7156	0.7335	0.7438	**0.7328**	0.7269	0.7080	5.00
	Hx	0.7065	0.7343	0.7320	0.7370	0.7308	0.7245	0.7156	5.85
	rCIM	0.7089	0.7222	0.7311	0.7420	0.7279	0.7244	0.7244	5.57
	Huber	0.7102	0.7242	0.7293	0.7462	0.7247	0.7236	0.7187	6.00
	Elastic	0.7066	0.7187	0.7318	0.7475	0.7232	0.7291	0.7215	5.71
	DANMF	0.7092	0.7157	0.7197	0.7182	0.7137	0.7157	0.7087	9.28
	DR-NMF	0.7168	0.7324	0.7299	0.7400	0.7243	0.7326	0.7298	4.71
	iDRNMF	0.7182	**0.7452**	**0.7379**	**0.7503**	0.7321	**0.7423**	**0.7380**	**1.28**
ACC	$L_{2,1}$	0.5580	0.5780	0.5795	0.5865	0.5735	0.5655	0.5405	6.21
	Fro	0.5585	0.5670	0.5865	0.5785	0.5660	0.5515	0.5440	7.57
	Cauchy	**0.5670**	0.5540	0.5885	0.5915	0.5725	0.5670	0.5385	5.57
	Hx	0.5455	0.5860	0.5775	0.5835	0.5770	0.5630	0.5525	6.50
	rCIM	0.5575	0.5710	**0.5895**	0.5910	0.5770	0.5730	0.5635	3.85
	Huber	0.5495	0.5725	0.5885	**0.5980**	0.5680	0.5645	0.5500	5.57
	Elastic	0.5475	0.5690	0.5795	0.5910	0.5745	0.5720	0.5505	5.78
	DANMF	0.5475	0.5800	0.5400	0.5700	0.5650	0.5675	0.5500	7.57
	DR-NMF	0.5645	0.5935	0.5780	0.5840	0.5610	0.5740	0.5645	4.78
	iDRNMF	0.5645	**0.6055**	0.5885	**0.5980**	**0.5790**	**0.5925**	**0.5810**	**1.57**

comparing its performance with other methods. The results of simulations for this noise model are comprehensively presented in Table 10. As anticipated, our iDRNMF shows a very good performance, and if we pay attention to the average rank obtained, although it has decreased compared to the noise distributions in the Ω set, it still has a privileged position compared to other methods. Although the DR-NMF model includes the Kullback–Leibler loss that matches the Poisson distribution, the proposed model achieves nearly identical results. This aligns with the expected behavior, reinforcing the adaptability of our model even in scenarios with noise distributions beyond the originally considered set Ω.

4.4. Analysis of iDRNMF terms contribution

The proposed objective function is designed as a weighted sum multi-objective cost function, where each loss has a significant impact on the final results. The model aims to minimize the worst-case expected loss function in each iteration. Consequently, the iDRNMF model increases the coefficient of the largest loss to pay more attention to it.

Table 5
The comparison results on the COIL20 dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G + L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.7625	0.7590	0.7361	0.7227	0.7367	0.7275	0.7266	6.86
	Fro	**0.7820**	0.7494	0.7281	0.7456	0.7426	0.7478	0.7651	4.57
	Cauchy	0.7658	0.7034	0.7527	0.7284	0.7075	0.7274	0.7831	5.57
	Hx	0.7650	0.7715	0.7331	0.7528	0.7524	0.7543	0.7516	3.71
	rCIM	0.7537	0.7159	0.7398	0.7437	0.7572	0.7527	0.7499	5.29
	Huber	0.7555	0.7341	0.7048	0.7383	0.7490	0.7572	0.7300	6.14
	Elastic	0.7136	0.7368	0.7244	0.7302	0.7500	0.7529	0.7571	6.29
	DANMF	0.7470	0.7572	0.7377	0.7380	0.7219	0.7284	0.7513	6.29
	DR-NMF	0.7365	0.6929	0.7041	0.7193	0.7045	0.7548	0.7459	8.57
	iDRNMF	0.7752	**0.7935**	**0.7583**	**0.7607**	**0.7753**	**0.7593**	**0.7903**	**1.14**
ACC	$L_{2,1}$	0.6986	0.6965	0.6562	0.6569	0.6514	0.6521	0.6347	6.14
	Fro	**0.7215**	0.6660	0.6326	0.6681	0.6611	0.6583	0.7069	5.00
	Cauchy	0.6736	0.6090	0.6771	0.6715	0.6285	0.6472	0.6986	6.07
	Hx	0.6785	0.6972	0.6556	**0.6861**	0.6729	0.6910	0.6951	3.50
	rCIM	0.6639	0.6313	0.6549	0.6701	0.6438	0.6917	0.6861	6.43
	Huber	0.6785	0.6715	0.6111	0.6715	0.6646	0.6951	0.6396	5.14
	Elastic	0.6361	0.6528	0.6611	0.6583	0.6597	0.6854	0.6972	6.00
	DANMF	0.6701	0.6701	0.6569	0.6722	0.6409	0.6340	0.6722	6.43
	DR-NMF	0.6417	0.6062	0.6083	0.6389	0.6201	0.6764	0.6729	8.86
	iDRNMF	0.6979	**0.7368**	**0.6806**	0.6799	**0.6882**	**0.7042**	**0.7188**	**1.43**

Table 6
The comparison results on the MNIST dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G + L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.3605	0.3050	0.3902	0.3393	0.3375	0.3606	0.3501	5.86
	Fro	**0.4013**	0.3535	0.3912	0.3765	0.3349	0.3323	0.3393	5.57
	Cauchy	0.3546	0.3611	0.3444	0.3191	**0.3655**	0.3443	**0.3958**	4.93
	Hx	0.3471	0.3639	0.3746	0.3461	0.3740	0.3349	0.3533	5.14
	rCIM	0.3491	0.3540	0.4032	0.3833	0.3527	0.3132	0.3466	5.43
	Huber	0.3364	0.3151	0.3444	0.3229	0.3366	0.3139	0.3522	7.79
	Elastic	0.3566	0.3162	0.4022	0.3536	0.3314	0.3566	0.3625	4.14
	DANMF	0.3610	0.3543	0.3987	0.3617	0.3622	0.3439	0.3643	3.86
	DR-NMF	0.1175	0.2121	0.2441	0.2363	0.2118	0.1743	0.1626	10.0
	iDRNMF	0.3897	**0.3644**	**0.4041**	**0.3888**	**0.3805**	**0.3625**	0.3942	**1.29**
ACC	$L_{2,1}$	0.4570	0.3780	0.4840	0.4300	0.4280	0.4340	0.4590	5.79
	Fro	0.4760	0.4350	0.4950	0.4700	0.4400	0.4030	0.4130	5.00
	Cauchy	0.4470	0.4310	0.4300	0.4120	0.4720	0.4400	0.4880	5.79
	Hx	0.4510	0.4420	0.4650	0.4300	0.4720	0.4440	0.4510	4.71
	rCIM	0.4210	0.4340	0.5030	0.4840	0.4730	0.3850	0.4430	5.00
	Huber	0.4280	0.3770	0.4400	0.4140	0.4250	0.4120	0.4460	7.71
	Elastic	0.4590	0.4130	0.4980	0.4570	0.4200	0.4480	0.4670	4.43
	DANMF	0.4540	0.4240	0.4800	0.4440	0.4520	0.4420	0.4350	5.57
	DR-NMF	0.2370	0.3210	0.3420	0.3300	0.3030	0.2770	0.2630	10.0
	iDRNMF	**0.4840**	**0.4470**	**0.5050**	**0.4920**	**0.4790**	**0.4520**	**0.4950**	**1.00**

However, the remaining loss functions also have their specific impacts and act as regularization terms during optimization in various scenarios. To analyze and demonstrate the contribution of each loss for input samples during the learning process, we conducted the iDRNMF model on the Yale dataset contaminated by Gaussian, Laplacian, Cauchy, and their combinations, separately. Specifically, 40% of pixels of all samples were contaminated, and the results are the average of 5 runs. The first three rows of Table 11 illustrate the NMI and ACC for the basis objective functions individually. The subsequent three rows show the results when iDRNMF runs with two of the objective functions in combination, and finally, the complete iDRNMF results are presented in the last row. As observed, each term positively impacts some scenarios, and combining these terms leads to superior performance in most cases. For instance, when the noise is Gaussian, the cases that incorporate the Frobenius norm in their multi-objective combination yield better results. This is also true for $L_{2,1}$ and Cauchy when iDRNMF faces Laplacian and Cauchy noises, respectively. Based on the findings, it is evident that each term of the proposed objective function plays a crucial role in achieving a good solution, and removing any one of them would likely lead to a decrease in average performance. It is worth noting that this experiment, as an ablation study, demonstrates the significance of each component within the proposed multi-objective cost function.

4.5. Robustness on various noise rates

In this section, we assess the robustness of iDRNMF through the application of a combination of noise (G+L+C) with varying rates (percentages of corrupted pixels) on the Yale dataset. The tasks can be challenging since they require filtering out plenty of outliers with large magnitudes to extract a clean subspace. The combinational noise is generated through a zero mean Gaussian distribution with a standard deviation of 5, a Zero Mean Laplacian distribution with a deviation parameter of 50, and Cauchy noise with a parameter $\gamma = 1$. The contamination levels range from 5% to 100%, representing the proportion of samples affected by noise. Fig. 2 depicts the NMI and ACC of the proposed iDRNMF method in comparison to alternative methods, respectively. It is evident from the figures that, as the noise rate increases, the iDRNMF method consistently exhibits superior clustering results when compared to alternative methodologies. This observation underscores the heightened robustness and resilience of the proposed

Table 7
The comparison results on the Fashion-MNIST dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G+L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.4376	0.4576	0.4539	0.4464	0.4488	0.4658	0.4272	4.57
	Fro	0.4590	0.4560	0.4348	0.4331	0.4427	0.4523	0.4375	6.00
	Cauchy	0.4456	0.4437	0.4313	0.4593	0.4525	0.4364	0.4514	5.43
	Hx	0.4446	0.4470	0.4476	0.4608	0.4557	0.4265	0.4487	4.71
	rCIM	0.4644	0.4670	0.4445	0.4432	0.4204	0.4531	0.4448	4.71
	Huber	0.4414	0.4410	0.4411	0.4351	0.4024	**0.4671**	0.4299	6.71
	Elastic	0.4345	0.4613	0.4515	0.4552	0.4484	0.4442	0.4242	5.57
	DANMF	0.4213	0.3684	0.4378	0.4145	0.4293	0.4207	0.4006	9.43
	DR-NMF	0.4277	0.4485	0.4499	0.4364	0.4390	0.4498	0.4295	6.71
	iDRNMF	**0.4700**	**0.4751**	**0.4572**	**0.4696**	**0.4616**	0.4603	**0.4547**	**1.29**
ACC	$L_{2,1}$	0.4670	0.5022	0.5082	0.5180	0.4850	0.5132	0.4658	4.64
	Fro	0.5008	0.5020	0.4872	0.5036	0.4850	0.4930	0.4984	5.50
	Cauchy	0.4826	0.4954	0.4752	0.5122	0.4980	0.4880	0.4970	6.29
	Hx	0.4862	0.4994	0.5008	**0.5354**	0.5116	0.4760	0.5102	4.29
	rCIM	**0.5050**	0.5010	0.4948	0.5140	0.4686	0.4952	0.4968	5.00
	Huber	0.4850	0.4916	0.4920	0.4970	0.4324	0.5094	0.4688	6.71
	Elastic	0.4708	0.5072	0.5074	0.5166	0.4946	0.4922	0.4604	5.14
	DANMF	0.4220	0.3830	0.4900	0.4320	0.4640	0.4780	0.4250	9.43
	DR-NMF	0.4718	0.4886	0.5056	0.4896	0.4868	0.4966	0.4588	6.71
	iDRNMF	0.5020	**0.5146**	**0.5160**	0.5294	**0.5154**	**0.5184**	**0.5278**	**1.29**

Table 8
The comparison results on the OrganA-MNIST dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G+L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.5611	0.6077	0.5833	0.5466	0.6054	0.6204	0.5824	5.00
	Fro	0.5911	0.6014	0.5987	0.6215	0.5155	0.5846	0.6032	5.14
	Cauchy	0.5557	0.6296	0.6088	0.6035	0.5418	0.5914	0.5791	5.57
	Hx	0.6162	0.5804	0.5599	0.5419	0.5612	0.5552	0.532	8.14
	rCIM	0.5819	0.5663	0.5616	0.5452	0.5766	0.6152	0.5710	7.28
	Huber	0.6095	0.5936	0.6103	0.6103	0.6004	0.5621	0.5714	5.14
	Elastic	0.6026	0.5993	0.5952	0.5773	0.5800	0.5819	0.5554	6.42
	DANMF	0.6149	0.5971	0.5526	0.5620	0.5848	0.5743	0.5905	6.14
	DR-NMF	0.6066	0.6094	0.5556	0.6163	0.5958	0.6149	0.5598	5.12
	iDRNMF	**0.6493**	**0.6302**	**0.6390**	**0.6287**	**0.6539**	**0.6372**	**0.6292**	**1.00**
ACC	$L_{2,1}$	0.6200	0.6598	0.6291	0.6022	0.6452	0.6783	0.6658	5.86
	Fro	0.6438	0.6487	0.6689	0.6820	0.5510	0.6484	0.6470	6.00
	Cauchy	0.5823	0.6790	0.6943	0.6579	0.5800	0.6436	0.6415	5.57
	Hx	0.6621	0.6345	0.6150	0.6171	0.6039	0.6267	0.6019	7.57
	rCIM	0.6233	0.6604	0.5690	0.5994	0.6381	0.6798	0.6187	7.00
	Huber	0.6655	0.6601	0.6761	0.6823	0.6661	0.6062	0.6154	5.29
	Elastic	0.6341	0.6650	0.6522	0.6435	0.6313	0.6162	0.5872	6.86
	DANMF	0.6729	0.6689	0.5826	0.6216	0.6773	0.6119	0.6556	5.14
	DR-NMF	0.6599	0.6780	0.5951	0.7014	0.6422	0.6780	0.6410	4.71
	iDRNMF	**0.7005**	**0.6835**	**0.7100**	**0.7086**	**0.7037**	**0.6909**	**0.6777**	**1.00**

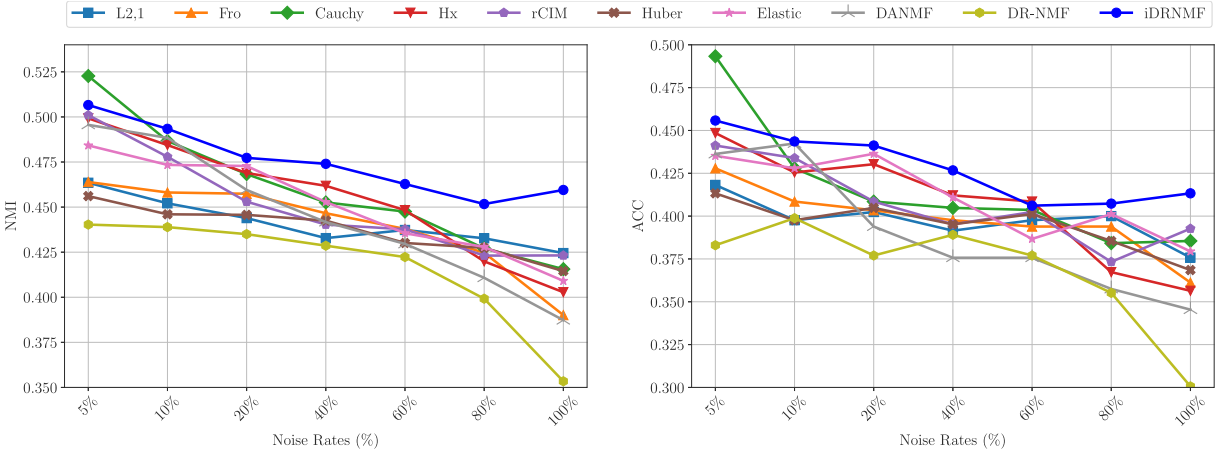


Fig. 2. NMI and ACC results on the Yale dataset with different combinational noise rates.

Table 9
The comparison results on the Blood-MNIST dataset for different types of noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

	Method	G	L	C	G+L	G+C	L+C	G+L+C	Avg Rank
NMI	$L_{2,1}$	0.3273	0.3459	0.3464	0.3473	0.3453	0.3250	0.3650	6.86
	Fro	0.3490	0.3399	0.3337	0.3498	0.3403	0.3434	0.3320	7.00
	Cauchy	0.3464	0.3430	0.3333	0.3486	0.3320	0.3443	0.3309	7.71
	Hx	0.3475	0.3570	0.3565	0.3550	0.3224	0.3538	0.3408	5.14
	rCIM	0.3647	0.3198	0.3305	0.3449	0.3434	0.3478	0.3602	6.71
	Huber	0.3403	0.3563	0.3441	0.3465	0.3584	0.3681	0.3596	5.57
	Elastic	0.3406	0.3665	0.3557	0.3492	0.3599	0.3475	0.3302	5.29
	DANMF	0.3515	0.3603	0.3205	0.3540	0.3495	0.3612	0.3521	4.71
	DR-NMF	0.3459	0.3597	0.3530	0.3228	0.3467	0.3634	0.3692	5.00
	iDRNMF	**0.3669**	**0.3742**	**0.3639**	**0.3605**	**0.3609**	**0.3722**	**0.3701**	**1.00**
ACC	$L_{2,1}$	0.4439	0.4900	0.4690	0.4567	0.4713	0.4655	0.5064	6.71
	Fro	0.4725	0.4439	0.4275	0.4742	0.4737	0.4696	0.4754	7.28
	Cauchy	0.4836	0.4707	0.4456	0.4778	0.4550	0.4760	0.4696	7.21
	Hx	0.4713	0.4924	0.5005	0.4702	0.4497	0.4935	0.4521	6.21
	rCIM	0.5011	0.4515	0.4456	0.4941	0.4678	0.4672	0.5023	6.07
	Huber	0.4713	0.4754	0.4853	0.4853	0.4877	0.5110	0.5029	4.35
	Elastic	0.4608	0.4853	0.4725	0.4900	0.4924	0.4889	0.4491	5.71
	DANMF	0.4959	0.4813	0.4515	0.4924	0.4734	0.5052	0.4637	5.00
	DR-NMF	0.4941	0.4830	0.4842	0.4456	0.4719	0.4982	0.4906	5.42
	iDRNMF	**0.5245**	**0.5029**	**0.5134**	**0.5040**	**0.5134**	**0.5140**	**0.5175**	**1.00**

Table 10
The comparison results on the Yale dataset for Poisson noise, evaluated based on NMI and ACC. The top three performances are highlighted in **boldface**, underlined, and double-underlined, respectively.

Noise level		$L_{2,1}$	Fro	Cauchy	Hx	rCIM	Huber	Elastic	DANMF	DR-NMF	iDRNMF
$\sigma = 5$	NMI	0.4877	0.4924	0.4897	0.4797	0.4813	0.4815	0.4864	0.4884	0.5082	**0.5193**
	ACC	0.4394	**0.4606**	0.4455	0.4424	0.4364	0.4364	0.4455	0.4424	0.4424	0.4576
$\sigma = 25$	NMI	0.4640	0.4501	0.4446	0.4653	0.4447	0.4432	0.4704	0.4595	**0.5011**	0.4852
	ACC	0.4273	0.4061	0.4000	0.4091	0.4061	0.4030	0.4303	0.3939	**0.4788**	0.4394
$\sigma = 50$	NMI	0.4600	0.4641	0.4606	0.4636	0.4683	0.4706	0.4517	0.4417	**0.4999**	0.4673
	ACC	0.4212	0.4152	0.4091	0.4182	0.4091	0.4273	0.3939	0.3757	**0.4545**	0.4364
$\sigma = 75$	NMI	0.4404	0.4522	0.4815	**0.4853**	0.4420	0.4422	0.4353	0.4428	0.4573	0.4689
	ACC	0.3970	0.3879	**0.4424**	**0.4424**	0.3879	0.3818	0.4030	0.4000	0.4091	0.4242
$\sigma = 100$	NMI	0.4566	0.4501	0.4489	0.4555	0.4323	0.4499	0.4425	0.4421	**0.4575**	0.4574
	ACC	0.4242	0.3939	0.4212	**0.4333**	0.3970	0.3939	0.4030	0.3818	0.4303	0.4000
Avg Rank	NMI	6.20	5.06	5.80	5.00	7.60	6.60	7.40	7.20	**1.80**	2.40
	ACC	5.20	6.10	5.10	3.70	7.80	7.80	5.10	8.40	**2.80**	3.00

Table 11
The comparison results on the Yale dataset for various noise, evaluated based on NMI and ACC.

Method	Gaussian		Laplacian		Cauchy		G+L+C	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
$L_{2,1}$	0.4330	0.3842	0.4136	0.3855	0.4104	0.3770	0.4245	0.3758
Fro	0.4071	0.3697	0.3987	0.3552	0.3928	0.3539	0.3901	0.3612
Cauchy	0.4240	0.3830	0.4258	0.3758	0.4311	0.3939	0.4156	0.3855
$\Omega = \{1, 2\}$	0.4465	0.4206	0.4433	0.4085	0.4240	0.3891	0.4520	0.4085
$\Omega = \{1, \text{Cau}\}$	0.4418	0.3988	**0.4595**	**0.4121**	0.4501	0.4061	0.4424	0.4048
$\Omega = \{2, \text{Cau}\}$	**0.4668**	**0.4303**	0.4188	0.3770	0.4488	0.4170	0.4345	0.4048
$\Omega = \{1, 2, \text{Cau}\}$	0.4507	0.4145	0.4518	0.4024	**0.4567**	**0.4242**	**0.4595**	**0.4133**

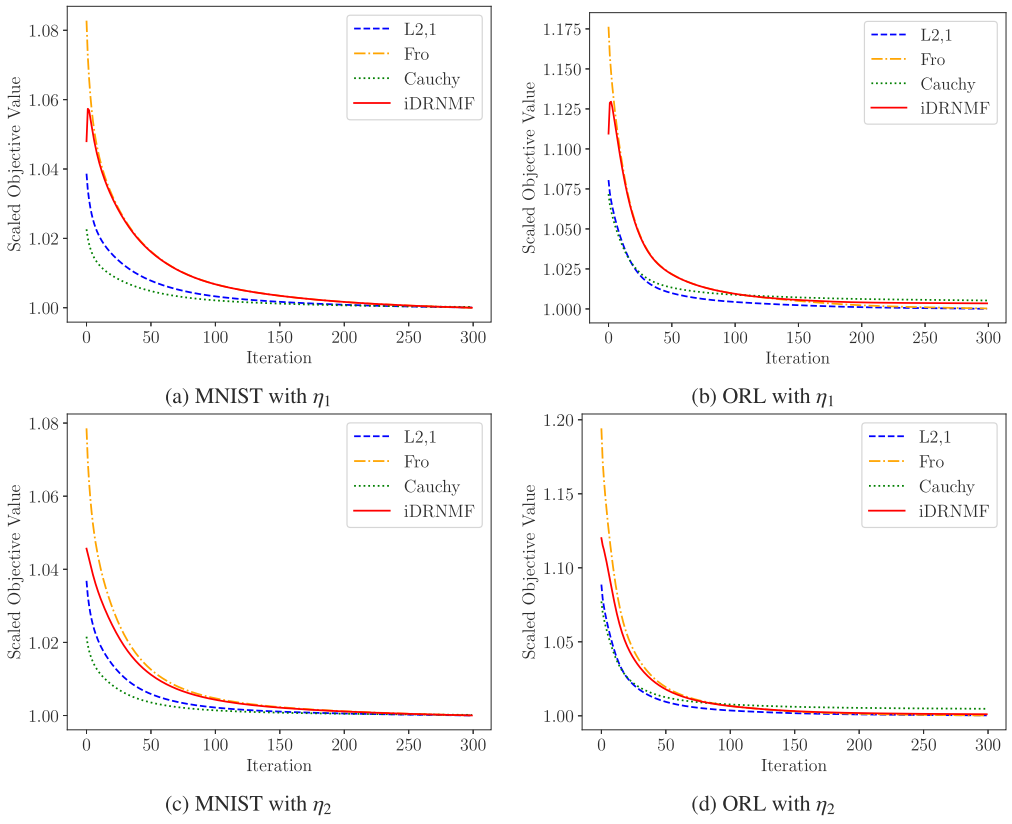
model, particularly in the context of the Yale dataset. The presented findings confirm the effectiveness of the iDRNMF method, particularly in higher ranges of noise. This establishes its superiority over the compared methods, affirming its suitability for tasks that require robust subspace extraction in the presence of diverse and challenging noise profiles.

4.6. Computational efficiency and runtime analysis

To provide a comprehensive comparison of computational efficiency, we report the running times of all evaluated methods across seven datasets in Table 12. The computational experiments were conducted using an Intel Core i7-3520M CPU with a 2.9 GHz clock speed and 8 GB of RAM. As a multi-objective and distributionally robust method with a computational complexity of $O(nmr)$, iDRNMF demonstrates competitive efficiency, particularly on Yale, ORL, MNIST, and Fashion. On these datasets, iDRNMF exhibits a moderate increase in computation time over Fro while significantly outperforming DR-NMF, another multi-objective method. Notably, iDRNMF remains consistently faster than Huber and DANMF across all datasets, reinforcing its efficiency advantage over these more computationally demanding approaches. For the high-dimensional datasets OrganA and Blood, iDRNMF continues to achieve a favorable balance between efficiency and robustness. While its runtime is higher than single-objective methods such as Fro, L21, and Cauchy, it remains significantly lower than DR-NMF, which requires considerably more computation time (79.34 s vs. 30.03 s on OrganA and 66.81 s vs. 24.19 s on Blood). Moreover, iDRNMF consistently outperforms Huber and DANMF, demonstrating

Table 12
Running time (in seconds) of various methods on different datasets.

Dataset	$L_{2,1}$	Fro	Cauchy	Hx	rCIM	Huber	Elastic	DANMF	DR-NMF	iDRNMF
Yale	0.156	0.125	0.156	0.172	0.171	0.563	0.156	1.736	0.578	0.187
ORL	0.906	0.719	0.797	0.797	0.906	1.736	0.969	4.319	3.079	1.094
COIL20	1.875	1.734	1.871	1.953	2.062	2.801	2.187	5.551	7.064	2.437
MNIST	0.578	0.491	0.547	0.656	0.719	1.025	0.672	3.658	2.234	0.75
Fashion	0.594	0.406	0.516	0.696	0.719	1.007	0.703	2.769	2.219	0.797
OrganA	19.865	14.954	20.91	20.751	24.392	34.29	24.399	43.599	79.34	30.034
Blood	16.661	11.953	17.969	17.894	20.972	30.257	19.44	37.584	66.809	24.192



(a) MNIST with η_1 (b) ORL with η_1

(c) MNIST with η_2 (d) ORL with η_2

Fig. 3. Convergence analysis of the iDRNMF model on the three datasets with different η.

superior scalability to high-dimensional settings. Given the additional complexity introduced by its distributionally robust formulation, these results highlight iDRNMF's ability to achieve enhanced robustness while maintaining feasible computation times. It is also important to note that both iDRNMF and DR-NMF involve an initialization phase for single-objective optimization, which is not separately reported in the table. Nevertheless, the overall runtime results confirm that iDRNMF strikes a practical balance between computational efficiency and robustness, making it a viable choice for high-dimensional and noisy data scenarios.

4.7. Convergence analysis

In this section, we explore experimental results to confirm the convergent property of the optimization algorithm. Fig. 3 illustrates the convergence of the iDRNMF model (depicted by the red line) across the ORL and MNIST datasets within the first 300 iterations. The y-axis represents the scaled value of objective functions, while the x-axis corresponds to the iteration numbers. The plots show that the objective value sequences on these datasets converge, providing empirical validation of the convergence feature of the proposed algorithm. Furthermore, our observations indicate that the proposed model typically converges within approximately 200 iterations, highlighting its rapid convergence and efficiency. The changes in the scaled values of the objective functions for $L_{2,1}$ (blue line), Frobenius (orange line), and the Cauchy (green line) during the updating of the proposed model are also depicted. Notably, the convergence of all basis cost functions is guaranteed simultaneously with the convergence of the main objective function. The objective approaches one, indicating that each objective is close to the objective obtained when minimizing a single objective. This convergence suggests that the algorithm is efficiently minimizing each objective function, resulting in an overall enhancement of the performance of the model.

In Section 3.3.2, we discussed the parameter η and its impact on convergence. This parameter determines the rate of change of λ_τ and should satisfy $0 < \eta < 1$. Larger values, equivalent to higher changes in the impact of basis objective functions, may lead to the dominance of the largest value of objective functions for some iterations. Conversely, smaller values prevent this dominance and provide more opportunities for other basis objective functions to influence the final residue. To evaluate the effect of η on the model's convergence, we experiment with different values. For this purpose, we define two momentum coefficients $\eta_1^{[t]} = \frac{1}{t+1}$ and $\eta_2^{[t]} = \frac{\lambda_{p^*}^{[t]}}{1+\lambda_{p^*}^{[t]}} \times \frac{1}{t+1}$ where t is the iteration number, p^* the index of the largest value among the basis objective functions, and $\lambda_{p^*}^{[t]}$ is the coefficient of the largest value among the basis objective functions in iteration t. Fig. 3 compares the convergence

behavior for both values, with the results for η_1 displayed on top and η_2 shown below. As observed, for η_1, the residue of the iDRNMF objective function initially rises in the early iterations and then decreases rapidly until convergence. This may be attributed to the higher value of the Frobenius residue and its dominance. In contrast, employing η_2, which generates smaller values, helps to alleviate the dominance of any single basis objective function, leading to a monotonically decreasing curve.

5. Conclusion and future works

This study introduces the instance-wise distributionally robust NMF model, adapted to handling various noise types concurrently. iDRNMF exhibits remarkable adaptability to various noise types, making it particularly valuable in fields where noise characteristics are unknown. This parameter-free method focuses on individual instances and leverages a multi-objective approach, resulting in superior robustness against diverse noise patterns. A key innovation of our work lies in the development of a unified iterative reweighted algorithm for model optimization. This approach ensures efficiency while maintaining accuracy in results. iDRNMF demonstrates the ability to handle common noise types like Gaussian, Laplacian, and Cauchy, making it well-suited for real-world applications with inherent noise. Extensive evaluations across nine established models on seven benchmark datasets validate the model's superior performance compared to existing methods. Additionally, we conducted tests using various noise models, including Gaussian, Laplacian, Cauchy, and combinations thereof, to comprehensively evaluate and compare the effectiveness of the proposed data representation model.

While iDRNMF shows resilience to various noise distributions, it faces challenges with extreme outliers. A promising direction for improvement is to enhance the model's ability to specifically address this issue. Inspired by truncated methods, we can integrate a robust thresholding mechanism to effectively filter out instances with unusually large residues. This approach will help identify and remove extreme outliers, thereby improving the model's resilience to anomalies. Additionally, incorporating faster optimization algorithms, such as Hierarchical Alternating Least Squares (HALS), with convergence guarantees, could further enhance the model's efficiency and reliability. Building upon the model's strength in handling diverse noise, another captivating avenue for future exploration lies in investigating its potential within a semi-supervised learning framework. In semi-supervised learning, unlabeled data often contains noise or irrelevant information. A distributionally robust model is less susceptible to the influence of noise, allowing it to extract meaningful patterns and information from the unlabeled data more effectively. Finally, our distributionally robust NMF model could be developed within real-world applications characterized by inherent data heterogeneity and the presence of diverse noise distributions. Particularly promising domains include medical imaging analysis, where images can be corrupted by a combination of artifacts such as Gaussian, Laplacian, and Cauchy.

CRediT authorship contribution statement

Wafa Barkhoda: Writing – original draft, Software, Methodology, Investigation, Data curation. **Amjad Seyedi:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Nicolas Gillis:** Writing – review & editing, Validation, Formal analysis. **Fardin Akhlaghian Tab:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Amjad Seyedi and Nicolas Gillis report financial support was provided by the European Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Amjad Seyedi acknowledges the support by the European Union (European Research Council (ERC) consolidator, eLinoR, no 1010 85607).

Data availability

Data will be made available on request.

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[2] C. Cheng, H. Li, J. Peng, W. Cui, L. Zhang, Deep high-order tensor convolutional sparse coding for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–11.
[3] N. Gillis, Nonnegative Matrix Factorization, SIAM, 2020.
[4] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[5] N.K. Logothetis, D.L. Sheinberg, Visual object recognition, Annu. Rev. Neurosci. 19 (1) (1996) 577—-621.
[6] C. Peng, P. Zhang, Y. Chen, Z. Kang, C. Chen, Q. Cheng, Fine-grained bipartite concept factorization for clustering, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 26254–26264.
[7] D. Wang, T. Li, P. Deng, F. Zhang, W. Huang, P. Zhang, J. Liu, A generalized deep learning clustering algorithm based on non-negative matrix factorization, ACM Trans. Knowl. Discov. Data (ISSN: 1556-4681) 17 (7) (2023).
[8] A. Mohammadi, S.A. Seyedi, F. Akhlaghian Tab, R. Pir Mohammadiani, Diverse joint nonnegative matrix tri-factorization for attributed graph clustering, Appl. Soft Comput. (ISSN: 1568-4946) 164 (2024) 112012.
[9] R. Mahmoodi, S.A. Seyedi, A. Abdollahpouri, F. Akhlaghian Tab, Enhancing link prediction through adversarial training in deep nonnegative matrix factorization, Eng. Appl. Artif. Intell. 133 (2024) 108641.
[10] C. Peng, Y. Zhang, Y. Chen, Z. Kang, C. Chen, Q. Cheng, Log-based sparse nonnegative matrix factorization for data representation, Knowl.-Based Syst. 251 (2022) 109127.
[11] C. Peng, Z. Zhang, Z. Kang, C. Chen, Q. Cheng, Nonnegative matrix factorization with local similarity learning, Inform. Sci. (ISSN: 0020-0255) 562 (2021) 325–346.
[12] H. Huang, G. Zhou, Q. Zhao, L. He, S. Xie, Comprehensive multiview representation learning via deep autoencoder-like nonnegative matrix factorization, IEEE Trans. Neural Networks Learn. Syst. 35 (5) (2024) 5953–5967.
[13] D. Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using L2,1-norm, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 673–682.
[14] Q. Ke, T. Kanade, Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, Vol. 1, 2005, pp. 739–746.
[15] L. Du, X. Li, Y.-D. Shen, Robust nonnegative matrix factorization via half-quadratic minimization, in: IEEE 12th International Conference on Data Mining, 2012, pp. 201–210.
[16] S. Peng, W. Ser, Z. Lin, B. Chen, Robust sparse nonnegative matrix factorization based on maximum correntropy criterion, in: IEEE International Symposium on Circuits and Systems, ISCAS, 2018, pp. 1–5.
[17] A. Hamza, D. Brady, Reconstruction of reflectance spectra using robust nonnegative matrix factorization, IEEE Trans. Signal Process. 54 (9) (2006) 3637–3642.
[18] S. Li, S. Wu, C. Tang, J. Zhang, Z. Wei, Robust nonnegative matrix factorization with self-initiated multigraph contrastive fusion, IEEE Trans. Neural Networks Learn. Syst. (2024) 1–15.
[19] A. Liutkus, D. Fitzgerald, R. Badeau, Cauchy nonnegative matrix factorization, in: IEEE Workshop on Applications of Signal Processing To Audio and Acoustics, WASPAA, 2015, pp. 1–5.
[20] S. Peng, B. Chen, L. Sun, W. Ser, Z. Lin, Constrained maximum correntropy adaptive filtering, Signal Process. 140 (2017) 116–126.
[21] H. Gao, F. Nie, W. Cai, 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.
[22] N. Guan, T. Liu, Y. Zhang, D. Tao, L.S. Davis, Truncated Cauchy non-negative matrix factorization, IEEE Trans. Pattern Anal. Mach. Intell. 41 (1) (2019) 246–259.
[23] H.E. Scarf, K. Arrow, S. Karlin, A Min-Max Solution of an Inventory Problem, Rand Corporation Santa Monica, 1957.

[24] N. Gillis, L. Hien, V. Leplat, V.F. Tan, Distributionally robust and multi-objective nonnegative matrix factorization, IEEE Trans. Pattern Anal. Mach. Intell. (ISSN: 1939-3539) 44 (08) (2022) 4052–4064.
[25] J.S. Cavazos, J.A. Fessler, L. Balzano, ALPCAH: Sample-wise heteroscedastic PCA with tail singular value regularization, in: International Conference on Sampling Theory and Applications (SampTA), 2023, pp. 1–6.
[26] L. Chen, C.-T. Wu, C.-H. Lin, R. Dai, C. Liu, R. Clarke, G. Yu, J.E. Van Eyk, D.M. Herrington, Y. Wang, swCAM: estimation of subtype-specific expressions in individual samples with unsupervised sample-wise deconvolution, Bioinformatics 38 (5) (2021) 1403—-1410.
[27] H. Wang, W. Yang, N. Guan, Cauchy sparse NMF with manifold regularization: A robust method for hyperspectral unmixing, Knowl.-Based Syst. 184 (2019) 104898.
[28] S. Ahn, J.H. Kim, V. Ramaswami, A new class of models for heavy tailed distributions in finance and insurance risk, Insurance Math. Econom. 51 (1) (2012) 43–52.
[29] Z. Liang, J. Wei, J. Zhao, H. Liu, B. Li, J. Shen, C. Zheng, The statistical meaning of kurtosis and its new application to identification of persons based on seismic signals, Sensors 8 (8) (2008) 5106–5119.
[30] S.A. Vavasis, On the complexity of nonnegative matrix factorization, SIAM J. Optim. 20 (3) (2010) 1364–1377.
[31] D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, Vol. 13, 2000.
[32] Q. Wang, X. He, X. Jiang, X. Li, Robust bi-stochastic graph regularized matrix factorization for data clustering, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 390–403.
[33] H. Xiong, D. Kong, F. Nie, Cauchy balanced nonnegative matrix factorization, Artif. Intell. Rev. 56 (10) (2023) 11867—-11903.
[34] I. Daubechies, R. DeVore, M. Fornasier, C.S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, Commun. Pure Appl. Math.: A J. Issued By Courant Inst. Math. Sci. 63 (1) (2010) 1–38.
[35] H. Choi, S. Choi, A. Katake, Y. Choe, Learning alpha-integration with partially-labeled data, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 2058–2061.
[36] M. Jaggi, Revisiting Frank-Wolfe: Projection-free sparse convex optimization, in: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, (1) 2013, pp. 427–435.
[37] A. Nemirovski, Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems, SIAM J. Optim. 15 (1) (2004) 229–251.
[38] H. Xiong, D. Kong, Elastic nonnegative matrix factorization, Pattern Recognit. 90 (2019) 464–475.
[39] N. Salahian, F. Akhlaghian Tab, S.A. Seyedi, J. Chavoshinejad, Deep autoencoder-like NMF with contrastive regularization and feature relationship preservation, Expert Syst. Appl. 214 (2023) 119051.
[40] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, B. Ni, MEDMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification, Sci. Data 10 (1) (2023) 41.

Wafa Barkhoda is a faculty member in the department of Information Technology at Kermanshah University of Technology, Iran. He is currently a Ph.D. student in artificial intelligence and robotics in the department of Computer Engineering, University of Kurdistan, Iran. Barkhoda received his M.S. degree in Computer Engineering from Kurdistan University in 2010. His research interests include machine learning, representation learning, unsupervised learning, robustness, and data clustering.

Amjad Seyedi is a Ph.D. student in Matrix Theory and Optimization in the Department of Mathematics and Operational Research, Faculte polytechnique, University of Mons, Mons, Belgium. He received his Master's in Artificial Intelligence from the Department of Computer Engineering at the University of Kurdistan in 2018. His work mainly focused on matrix factorization and low-rank approximation.

Nicolas Gillis is a Professor with the Department of Mathematics and Operational Research, Faculte polytechnique, University of Mons, Mons, Belgium. His research interests include optimization, numerical linear algebra, machine learning, signal processing, and data mining. He is currently serving as an Associate Editor of the IEEE Transactions on Signal Processing and of the SIAM Journal on Matrix Analysis and Applications.

Fardin Akhlaghian Tab is currently the associate professor of Computer engineering at the University of Kurdistan, Iran, and his research focuses on machine learning pattern recognition, and computer vision. He did his Ph.D. in Computer Vision at the University of Wollongong in 2005. He holds a master's degree from Tehran University of Tarbiat Modarres in 1992.