Adversarial Nonnegative Matrix Factorization

Lei Luo [1 2] **Yanfu Zhang** [2] **Heng Huang** [1 2]

Abstract

Nonnegative Matrix Factorization (NMF) has become an increasingly important research topic in machine learning. Despite all the practical success, most of existing NMF models are still vulnerable to adversarial attacks. To overcome this limitation, we propose a novel Adversarial NMF (ANMF) approach in which an adversary can exercise some control over the perturbed data generation process. Different from the traditional NMF models which focus on either the regular input or certain types of noise, our model considers potential test adversaries that are beyond the pre-defined constraints, which can cope with various noises (or perturbations). We formulate the proposed model as a bilevel optimization problem and use Alternating Direction Method of Multipliers (ADMM) to solve it with convergence analysis. Theoretically, the robustness analysis of ANMF is established under mild conditions dedicating asymptotically unbiased prediction. Extensive experiments verify that ANMF is robust to a broad categories of perturbations, and achieves state-of-the-art performances on distinct real-world benchmark datasets.

1. Introduction

The nonnegative matrix factorization (NMF) has been a prevalent nonnegative dimensionality reduction method and successfully applied to many fields such as feature extraction (Zhi et al., 2011), video tracking (Bucak & Gunsel, 2007), image processing (Geng et al., 2012) and document clustering (Guan et al., 2012). Given a data matrix $\mathbf{Y} \in \Re^{m\times N}$ with non-negative entries, the goal of NMF is to factorize $\mathbf{Y}$ into the product $\mathbf{AX}$ of two nonnegative matrices, with n columns in $\mathbf{A}$, where n is generally small.

Since NMF was popularized by Lee and Seung (Lee & Seung, 1999), various NMF methods have been proposed. However, most of them independently deal with each element of $\mathbf{X}$, regardless of the relationship between elements. To address this issue, some scholars considered the structural information of data or variate in modeling. Specifically, Cai *et al.* (Cai et al., 2010) exploited the intrinsic geometry of the data distribution and constructed a nearest neighbor graph to model the manifold structure term. Kim *et al.* (Kim et al., 2012) leveraged a mixed norm regularization to promote group sparsity in the factor matrices of NMF. Haeffele *et al.* (Haeffele et al., 2014) explored a matrix factorization technique suitable for large datasets that captures additional structure in the factors by using a projective tensor norm.

It is noteworthy that the NMF models mentioned above are limited to the regular input, and they are more inclined to use the simple L_2-norm to characterize the residual between matrices $\mathbf{Y}$ with $\mathbf{AX}$. Such a strategy obviously contradicts to the practical observations where data often contains noise which may have heavy-tailed attribute. Thus, in recent research, Robust Nonnegative Matrix Factorization (RNMF) methods have gained increasing attentions. For instance, Kong *et al.* (Kong et al., 2011) presented a novel robust formulation of NMF by using $L_{2,1}$-norm loss function which can accommodate outliers and noises in a better way than L_2-norm. Subsequently, Huang *et al.* (Huang et al., 2014) generalized (Kong et al., 2011) by adopting manifold regularization. Based on the correntropy induced metric, Du *et al.* (Du et al., 2012) introduced a robust NMF method which can effectively cope with the practical noise. Guan *et al.* (Guan et al., 2017) proposed a Truncated Cauchy NMF loss that handles outliers by truncating large errors, while Gao *et al.* (Gao et al., 2015) employed capped norm to remove the effect of extreme data outliers in NMF.

However, the RNMF methods are only suitable for some special types of noises, *e.g.*, Laplacian or Cauchy noise, which cannot show the flexibility in facing the worst-case (*i.e.*, adversarial) perturbations of data points. Compared with random noise, perturbations caused by specific features or the noise are more common and complex, but often contaminate the input data in practical applications. Sometimes, they are hardly perceptible to the human eye, yet sufficient to change the output of an algorithm. As a result, how to increase the robustness of models against the general

[1] JD Finance America Corporation, Mountain View, CA, USA [2] Department of Electrical and Computer Engineering, University of Pittsburgh, PA, USA. Correspondence to: Heng Huang <heng.huang@pitt.edu>.

Proceedings of the 37th International Conference on Machine Learning, Online, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

perturbations has become a very important task in NMF.

To address this challenging problem, in this paper, we introduce a novel Adversarial Nonnegative Matrix Factorization (ANMF) model by emphasizing potential test adversaries that are beyond the pre-defined constraints. Specifically, we leverage adversarial perturbations of $\mathbf{Y}$ to learn the adversarial feature matrix $\tilde{\mathbf{A}}$ data and weight matrix $\mathbf{X}$. Differing from the traditional NMF models which either focus on the regular data point or use the simple matrix norm to characterize error between matrices $\mathbf{AX}$ and $\mathbf{Y}$, our model fully utilizes an adversary of input data $\mathbf{Y}$ to exercise some control over the data generation process to improve the stability of $\mathbf{A}$. Thus, it does not rely on any noise assumption. The proposed model is formulated as a bilevel optimization problem and an efficient optimization algorithm is derived to solve it with the convergence analysis. In addition, we establish the complete theoretical guarantees for ANMF under mild conditions. These results soundly support the rationality of ANMF. The main contributions of this paper are summarized as follows:

- From learner and attacker perspectives, we propose a novel Adversarial Nonnegative Matrix Factorization (ANMF) model which can handle different types of noise or perturbations.
- The robustness analysis is provided under different conditions, which theoretically guarantees the soundness of our model.
- Alternating Direction Method of Multipliers is applied to solving ANMF, where we can achieve the closed-form solution for each sub-problem.
- Empirical studies are performed on real-world benchmark data sets with various noise conditions. All results demonstrate that the proposed algorithm can consistently outperform other related methods.

Notations: Throughout this paper, the bold capital and bold lowercase symbols are used to represent matrices and vectors, respectively. If all elements of a matrix $\mathbf{A}$ are greater than or equal to 0, we denote it by $\mathbf{A} \geq 0$. $||\mathbf{A}||_F$ and $||\mathbf{A}||_{2,1}$ mean Frobenius norm and $L_{2,1}$-norm of the matrix $\mathbf{A}$, respectively. Finally, a $p \times p$-identity matrix is denoted by $\mathbf{I}_p$ and $\mathbf{0}$ denotes is a zero matrix.

2. Backgrounds

Nonnegative matrix factorization. The standard NMF models factorize an observation matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in \Re^{m\times N}$ as two nonnegative factors $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n] \in \Re^{m\times n}$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdot\cdot\cdot, \mathbf{x}_N] \in \Re^{n\times N}$ such that $\mathbf{Y} \approx \mathbf{AX}$. $\mathbf{A}$ is often referred to as feature matrix and $\mathbf{X}$ referred as weights. It has been shown that the non-negativity constraint on the coefficients forcing features to combine, but not cancel out, can lead to much more interpretable features and improved downstream performance of the learned features. The standard NMF can be formulated as follows:

$$\min_{\mathbf{A},\mathbf{X}} \parallel \mathbf{Y} - \mathbf{AX} \parallel_F^2, s.t., \mathbf{A} \geq 0,\ \mathbf{X} \geq 0. \tag{1}$$

Numerous algorithms have been developed in the literature for finding its high-quality solutions. The most representative algorithm is multiplicative update which alternates between solving certain surrogate functions for $\mathbf{A}$ and $\mathbf{X}$, respectively (Gonzalez & Zhang, 2005). The Alternating Nonnegative Least Square (Lin, 2007) is another class of useful algorithms, which includes the projected gradient descent method (Hajinezhad et al., 2016).

Since the L_2-norm is sensitive to the practical noises, intensive efforts have been put to design robust NMF models. The common strategy of these methods is to adopt a special loss function to characterize the errors between $\mathbf{Y}$ and $\mathbf{AX}$, *e.g.*, $L_{2,1}$-norm (Kong et al., 2011), Capped norm (Gao et al., 2015), Correntropy induced metric (Du et al., 2012) and Truncated Cauchy function (Guan et al., 2017).

Adversarial Perturbations. State-of-the-art machine learning models have achieved high accuracy on a broad range of datasets, yet can be easily misled by small perturbations of their input. While such perturbations may be simple noise to a human or even imperceptible, they often cause modern models to misclassify their input with high confidence. To provide adversarial robustness against the adversarial attack, a standard technique, which is called adversarial training (Farnia et al., 2018), follows empirical risk minimization training over the adversarially-perturbed samples by solving

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} l(f_{\mathbf{w}}(\mathbf{v}_i + \Delta_{\mathbf{w}}(\mathbf{v}_i)), \mathbf{b}_i), \tag{2}$$

where $l(\cdot,\cdot)$ is a loss function, $f_{\mathbf{w}}(\cdot)$ is the output function, $\Delta_{\mathbf{w}}(\mathbf{v}_i)$ is the adversarial additive perturbation (or noise) for sample $\mathbf{v}_i$ and each $\mathbf{b}_i$ is a test sample or the label vector.

3. Adversarial Nonnegative Matrix Factorization

Model (1) can be further written as:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^{N} \parallel \mathbf{y}_i - \mathbf{A}\mathbf{x}_i \parallel_2^2, \\ &s.t.,\ \mathbf{A} \geq 0,\ \mathbf{x}_i \geq 0, i = 1, 2, \cdots, N. \end{aligned} \tag{3}$$

Although some robust NMF methods (Kong et al., 2011; Gao et al., 2015; Du et al., 2012) may be effective for handling practical noise, their aim is to search for an appropriate loss function to characterize the errors with special properties such that outlying data have a relatively smaller influence in the process of learning variables. Thus, the given loss function is only suitable for some specific noise. For other types of noises, it may be ineffective. Additionally, these robust NMF methods highly rely on the independence assumption of noise pixels, which is unrealistic for some structured noises caused by occlusions or illumination.

More recently, several efforts have focused on proactive approaches of modeling the learner and adversary as players in a game in which the learner chooses a classifier or a learning algorithm, and the attacker modifies either the training or test data (Li & Vorobeychik, 2014; Großhans et al., 2013; Tong et al., 2018). These methods have shown great potential when handling data with adversarial perturbations. In this paper, we consider the data perturbed by noises as the attacker and present an adversarial version of (3) to improve the performance of NMF, where our main task is to model the adversary's attack strategy and develop robust learning models to mitigate the attack.

For the convenience, we assume that the learned feature data $\mathbf{A}$ and given data $\mathbf{Y}$ are drawn from an unknown distribution $\mathcal{D}$ at training time. By contrast, at application time, the test data can be generated either from $\mathcal{D}$, the same distribution as the training data, or from $\tilde{\mathcal{D}}$, a modification of $\mathcal{D}$ generated by an attacker. The action of the learner is to select parameters $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ of the Eq. (3). It is assumed that the attacker has an instance-specific target, and encourages that the prediction made by learner on the modified instance, $\mathbf{y}_i = \tilde{\mathbf{A}}\mathbf{x}_i$ $(i = 1, \cdots, N)$, is close to this target.

The cost functions of each learner (Cl) and the attacker (Ca) are estimated by:

$$Cl(\mathbf{X}, \tilde{\mathbf{A}}, \mathbf{A}) = \alpha G(\tilde{\mathbf{A}}\mathbf{X}, \mathbf{Y}) + \beta G(\mathbf{A}\mathbf{X}, \mathbf{Y}) \tag{4}$$

and

$$Ca(\mathbf{X}, \tilde{\mathbf{A}}, \mathbf{A}) = G(\tilde{\mathbf{A}}\mathbf{X}, \mathbf{Z}) + \lambda G(\tilde{\mathbf{A}}, \mathbf{A}). \tag{5}$$

Here parameters $\alpha, \beta, \lambda > 0$, $G(\cdot, \cdot)$ is a given metric and $\mathbf{Z} = \mathbf{Z}(\mathbf{Y})$ is the instance-specific target for attacker.

Ultimately, our model is expressed as:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{A}} Cl(\mathbf{X}, \tilde{\mathbf{A}}^*(\mathbf{X}), \mathbf{A}) \qquad\qquad (6)\\ \textit{s.t.} \quad &\tilde{\mathbf{A}}^*(\mathbf{X}) = \arg\min_{\tilde{\mathbf{A}}} Ca(\mathbf{X}, \tilde{\mathbf{A}}, \mathbf{A}), \mathbf{X} \geq 0, \mathbf{A} \geq 0. \end{aligned}$$

Our method provides a general framework for dealing with practical perturbations which may be caused by Laplace, Gaussian or structured noises. It does not depend on any assumption on perturbations (or noises). Thus, compared with those robust methods, model (6) is more adaptive for the practical NMF problems.

Remark 1. In general, $\mathbf{Z}$ can be set using two approaches: (a) $\mathbf{Z} = \mathbf{Y} + t\mathbf{1}$, where t is sampled from $[0, 5\sigma_r]$ and σ_r is the standard deviation of test image samples $[\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]$, $\mathbf{1}$ is a matrix with all elements equal to one; (b) $\mathbf{Z} = \mathbf{Y} + \triangle$, where $\triangle$ is the practical noises caused by outlier, illuminations or occlusions.

Remark 2. We use manual perturbations in model (6). In fact, we can impose some priors on perturbations to automatically learn them. But here we omit such a process.

Model (6) is factually a bilevel optimization problem. Specially, if the feature matrix $\mathbf{A}$ is fixed, it can be considered as the matrix version of the Stackelberg Equilibrium problem (Tong et al., 2018) (More details can be seen in the supplementary materials). In this paper, we set $G(\tilde{\mathbf{A}}, \mathbf{A}) = \| \tilde{\mathbf{A}} - \mathbf{A} \|_F^2$. Then, for the lower level, we have the following closed-form solution:

Theorem 1. Given $\mathbf{X}$, the best response of the attacker is

$$\tilde{\mathbf{A}}^*(\mathbf{X}) = (\lambda\mathbf{A} + \mathbf{Z}\mathbf{X}^T)(\lambda\mathbf{I}_n + \mathbf{X}\mathbf{X}^T)^{-1}. \tag{7}$$

Since there is an inverse of complicated matrix in (7), it is difficult to solve problem (6) by directly substituting (7) into (6). To mitigate this limitation, we consider (7) as a constraint of (6), which leads to the following problem:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{A},\tilde{\mathbf{A}}} Cl(\mathbf{X}, \mathbf{A}, \tilde{\mathbf{A}}), \qquad\qquad (8)\\ \textit{s.t.} \quad &\tilde{\mathbf{A}}(\lambda\mathbf{I} + \mathbf{X}\mathbf{X}^T) - (\lambda\mathbf{A} + \mathbf{Z}\mathbf{X}^T) = 0, \mathbf{X} \geq 0, \mathbf{A} \geq 0. \end{aligned}$$

Let $\varphi(\tilde{\mathbf{A}}, \mathbf{X}) = \tilde{\mathbf{A}}(\lambda\mathbf{I} + \mathbf{X}\mathbf{X}^T) - (\lambda\mathbf{A} + \mathbf{Z}\mathbf{X}^T)$. Problem (8) can be approximated as:

$$\min_{\mathbf{X},\mathbf{A},\tilde{\mathbf{A}}} Cl(\mathbf{X}, \mathbf{A}, \tilde{\mathbf{A}}) + \gamma\|\varphi(\tilde{\mathbf{A}}, \mathbf{X})\|_F^2, \ \textit{s.t.}, \mathbf{X} \geq 0, \mathbf{A} \geq 0. \tag{9}$$

Thus, we focus on model (9) in this paper.

4. Theoretical Analysis

We define the empirical reconstruction error of NMF as follows:

$$R_N(\mathbf{A}) = \frac{1}{N} \sum_{i=1}^{N} \min_{\mathbf{x}} ||\mathbf{y}_i - \mathbf{A}\mathbf{x}||_2^2. \tag{10}$$

Denote the expectation operator by $\mathcal{E}$. Then, the expected reconstruction error of ANMF can be written as

$$R(\mathbf{A}) = \mathcal{E}_y R_N(\mathbf{A}). \tag{11}$$

The Rademacher complexity is defined as:

$$\mathfrak{R}(F) = \mathcal{E}_{\sigma,x} \sup_{f \in F} \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \tag{12}$$

$\sigma_1, \cdots, \sigma_n$ are independent Rademacher variables, $f(\cdot)$ belongs to the $[a, b]$-value function class on $\mathbb{X}$ and $\mathbf{x} =$

$(x_1, x_2, \cdots, x_N)^T \in \mathbb{R}^N$ are independent and identically distributed examples.

Based on the above theorems and definitions, we can obtain the generalization bound of ANMF.

Theorem 2. For ANMF problem, assume that $\mathbf{Y}$ is upper bound by 1. For any learned normalized $\mathbf{A}$ and any $\delta > 0$, with probability at least $1 - \delta$, we have

$$|R(\mathbf{A}) - R_N(\mathbf{A})| \leq \min\left\{ \frac{14n\sqrt{n}}{\sqrt{N}} + \sqrt{\frac{n^2 \ln(16Nn)}{4N}} + \sqrt{\frac{\ln 2/\delta}{2N}}, \frac{2}{N} + \sqrt{\frac{mn \ln(4(1+n)\sqrt{m}nN) - \ln\frac{\delta}{2}}{2N}} \right\}. \tag{13}$$

It is easy to see that standard NMF problem (1) is a special case ($\alpha, \gamma = 0$) of ANMF. Similar to (Liu et al., 2016), we can obtain a dimensionality independent generalization bound for standard NMF problem (1) when the feature matrx $\mathbf{A}$ is orthogonal, *i.e.*,

Theorem 3. For orthogonal NMF problem, assume that $\mathbf{Y}$ upper bounded by 1. For any learned normalized $\mathbf{A}$ with and any $\delta > 0$, with probability at least $1 - \delta$, we have

$$|R(\mathbf{A}) - R_N(\mathbf{A})| \leq 6n\sqrt{\frac{\pi}{N}} + \sqrt{\frac{\ln 2/\delta}{2N}}. \tag{14}$$

Both Theorems 2 and 3 point out the asymptotically unbiased prediction of NMF methods. Specifically, Theorem 2 provides a slightly weaker but meaningful generalization bound for ANMF. Although Theorem 3 depends on the mild orthogonality assumption, its explicit representation can reflect more clearly the effectiveness of orthogonal ANMF than Theorem 2. All proofs are in supplementary materials.

5. Optimization Algorithm

In this section, we apply the Alternating Direction Method of Multipliers (ADMM) optimization algorithm to solve problem (9). The ADMM or the Augmented Lagrange Multipliers (ALM) method was presented originally in (Gabay & Mercier, 1975) and (Fortin & Glowinski, 2000), which has been extensively studied in the theoretical frameworks of Lagrangian functions (Fortin & Glowinski, 2000). Recently, it has been shown that ADMM are efficient for many convex and nonconvex programming problems arising from various applications (Chartrand & Wohlberg, 2013; Wang et al., 2015; Mei et al., 2018).

It should be noted that the iterative scheme of ADMM integrates the Gaussian-Seidel decomposition into iterations of the ALM in (Wang et al., 2015), which implies that the functions with regard to different variables are treated individually. Accordingly, the easier sub-problems could be generated. For the splitting, several auxiliary variables are introduced, and (9) is transformed into the following equivalent form:

$$\begin{aligned} \min_{\mathbf{H},\mathbf{J},\mathbf{U},\mathbf{X},\mathbf{A},\tilde{\mathbf{A}},\mathbf{B},\tilde{\mathbf{B}}} \ & \alpha\|\mathbf{H} - \mathbf{Y}\|_F^2 + \beta\|\mathbf{J} \ - \mathbf{Y}\|_F^2 \\ & + \gamma\|\mathbf{H}\mathbf{U} + \lambda\tilde{\mathbf{A}} - \lambda\mathbf{A} - \mathbf{Z}\mathbf{U}\|_F^2, \\ s.t. \ \ & \tilde{\mathbf{A}}\mathbf{X} = \mathbf{H}, \mathbf{A}\mathbf{X} = \mathbf{J}, \mathbf{X}^T = \mathbf{U}, \tilde{\mathbf{A}} = \tilde{\mathbf{B}}, \\ & \mathbf{A} = \mathbf{B}, \mathbf{B} \ \geq 0, \tilde{\mathbf{B}} \ \geq 0, \mathbf{U} \ \geq 0. \end{aligned} \tag{15}$$

Algorithm 1 Solving Eq. (9) via ADMM

```
**Input:** $\mathbf{Y} \in \mathbb{R}^{m \times N}$ and instance-specific target $\mathbf{Z}$
**Output:** feature matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and weight matrix $\mathbf{X} \in \mathbb{R}^{n \times N}$.
**Initialization:** $\tilde{\mathbf{A}}$ and $\mathbf{X}$ using the traditional $K$-means method, $\mathbf{A}^0 = \mathrm{A}_K$, where $\mathbf{A}_K$ is the clustering centroid obtained by $K$-means method, $\mathbf{X}^0 = \mathbf{X}_K + 0.2$, where $\mathbf{X}_K$ is the $K$-means clustering result. $\mathbf{U}^0 = \mathbf{X}^{0^T}$, $\tilde{\mathbf{B}}^0 = \tilde{\mathbf{A}}^0 = \mathbf{A}^0$, $\mathbf{M}^0 = \mathbf{0}$.
**repeat**
    Update $(\mathbf{H}, \mathbf{B}, \mathbf{J})$ by
        $\mathbf{H}^{k+1} \leftarrow \mathbf{R}_1(2\alpha\mathbf{I}_N + \mu\mathbf{I}_N + 2\gamma\mathbf{U}^k\mathbf{U}^{k^T})^{-1}$,
        $\tilde{\mathbf{B}}^{k+1} \leftarrow \max(0, \tilde{\mathbf{A}} + \frac{1}{\mu}\mathbf{M}_1^k)$, $\mathbf{J}^{k+1} \leftarrow \frac{\mathbf{R}_2}{2\beta + \mu}$,
    Update $(\mathbf{U}, \tilde{\mathbf{B}})$ by
        $\mathbf{U}^{k+1} \leftarrow \max(0, (2\gamma(\mathbf{H}^{k+1} - \mathbf{Z})^T$
            $\cdot (\mathbf{H}^{k+1} - \mathbf{Z}) + \mu\mathbf{I}_N)^{-1}\mathbf{R}_3)$,
        $\tilde{\mathbf{B}}^{k+1} \leftarrow \max(0, \mathbf{A}^k + \frac{1}{\mu}\mathbf{M}_1^k)$,
    Update $\mathbf{A}$ by
        $\mathbf{A}^{k+1} \leftarrow \mathbf{R}_4(2\gamma\lambda^2\mathbf{I}_n + \mu\mathbf{I}_n + \mathbf{X}^k\mathbf{X}^{k^T})^{-1}$,
    Update $\tilde{\mathbf{A}}$ by
        $\tilde{\mathbf{A}}^{k+1} \leftarrow \mathbf{R}_5(2\gamma\lambda^2\mathbf{I}_n + \mu\mathbf{I}_n + \mathbf{X}^k\mathbf{X}^{k^T})^{-1}$,
    Update $\mathbf{X}$ by
        $\mathbf{X}^{k+1} \leftarrow ((\tilde{\mathbf{A}}^{k+1})^T\tilde{\mathbf{A}}^{k+1} + \mathbf{A}^{k+1^T}\mathbf{A}^{k+1} + \mathbf{I}_n)^{-1}\mathbf{R}_6$;
    Update $\mathbf{M}$ by
        $\mathbf{M}^{k+1} \leftarrow \mathbf{M}^k + \mu(\phi(\tilde{\mathbf{A}}^{k+1}, \mathbf{A}^{k+1})\psi(\mathbf{X}^{k+1})$
            $- \nu(\mathbf{H}^{\tilde{k}+1}, \mathbf{J}^{k+1}, \tilde{\mathbf{B}}^{k+1}, \mathbf{B}^{k+1}, \mathbf{U}^{k+1}))$.
**until** Converge
```

Let us write the augmented Lagrange function for the problem (15) as :

$$\begin{aligned} & L_\mu(\mathbf{H}, \mathbf{B}, \mathbf{J}, \mathbf{U}, \tilde{\mathbf{B}}, \mathbf{A}, \tilde{\mathbf{A}}, \mathbf{X}, \mathbf{M}) \\ = & \alpha\|\mathbf{H} - \mathbf{Y}\|_F^2 + \beta\|\mathbf{J} \ - \mathbf{Y}\|_F^2 \\ & + \gamma\|\mathbf{H}\mathbf{U} + \lambda\tilde{\mathbf{A}} - \lambda\mathbf{A} - \mathbf{Z}\mathbf{U}\|_F^2 + \frac{\mu}{2}(\|\tilde{\mathbf{A}} - \tilde{\mathbf{B}} + \frac{1}{\mu}\mathbf{M}_1\|_F^2 \\ & + \|\mathbf{A} - \mathbf{B} + \frac{1}{\mu}\mathbf{M}_2\|_F^2 + \|\tilde{\mathbf{A}}\mathbf{X} - \mathbf{H} + \frac{1}{\mu}\mathbf{M}_3\|_F^2 \\ & + \|\mathbf{A}\mathbf{X} - \mathbf{J} + \frac{1}{\mu}\mathbf{M}_4\|_F^2 + \|\mathbf{X}^T - \mathbf{U} + \frac{1}{\mu}\mathbf{M}_5\|_F^2), \end{aligned} \tag{16}$$

where $\mu > 0$ is a tunable penalty parameter.

Then, ADMM is applied to minimizing the augmented Lagrangian problem (16) with respect to $\mathbf{H}, \mathbf{J}, \mathbf{U}, \mathbf{X}, \mathbf{A}, \tilde{\mathbf{A}}, \mathbf{B}, \tilde{\mathbf{B}}$ alternately.

The iterative scheme of ADMM for problem (15) is summarized in Algorithm 1, where

$$\begin{aligned} \phi(\tilde{\mathbf{A}}, \mathbf{A}) &= \mathrm{diag}(\tilde{\mathbf{A}}, \mathbf{A}, \tilde{\mathbf{A}}, \mathbf{A}, \mathbf{I}_N), \\ \psi(\mathbf{X}) &= \mathrm{diag}(\mathbf{X}, \mathbf{X}, \mathbf{I}_n, \mathbf{I}_n, \mathbf{X}^T), \end{aligned} \tag{17}$$

$$\begin{aligned} \nu(\tilde{\mathbf{H}}, \mathbf{J}, \tilde{\mathbf{B}}, \mathbf{B}, \mathbf{U}) &= \mathrm{diag}(\mathbf{H}, \mathbf{J}, \tilde{\mathbf{B}}, \mathbf{B}, \mathbf{U}), \\ \mathbf{M} &= \mathrm{diag}(\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4, \mathbf{M}_5), \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{R}_1 &= 2\alpha\mathbf{Y} + 2\gamma(\lambda\mathbf{A}^k + \mathbf{Z}\mathbf{U}^k - \lambda\tilde{\mathbf{A}}^k){\mathbf{U}^k}^T \\ & + \mu(\tilde{\mathbf{A}}^k\mathbf{X}^k + \frac{1}{\mu}\mathbf{M}_3^k), \\ \mathbf{R}_2 &= 2\beta\mathbf{Y} + \mu(\tilde{\mathbf{A}}^k\mathbf{X}^k + \frac{1}{\mu}\mathbf{M}_3^k), \end{aligned} \tag{19}$$

$$\mathbf{R}_3 = 2\gamma(\mathbf{H}^{k+1} - \mathbf{Z})^T(\lambda\mathbf{A}^k - \lambda\tilde{\mathbf{A}}^k) + \mu({\mathbf{X}^k}^T + \frac{1}{\mu}\mathbf{M}_5^k), \tag{20}$$

$$\begin{aligned} \mathbf{R}_4 =& 2\gamma\lambda(\mathbf{H}^{k+1}\mathbf{U}^{k+1} + \lambda\tilde{\mathbf{A}}^k - \mathbf{Z}\mathbf{U}^{k+1}) \\ & + \mu(\mathbf{B}^{k+1} - \frac{1}{\mu}\mathbf{M}_2^k) + \mu(\mathbf{J}^{k+1} - \frac{1}{\mu}\mathbf{M}_4^k){\mathbf{X}^k}^T, \end{aligned} \tag{21}$$

$$\begin{aligned} \mathbf{R}_5 =& 2\gamma\lambda(\lambda\mathbf{A}^{k+1} + \mathbf{Z}\mathbf{U}^{k+1} - \mathbf{H}^{k+1}\mathbf{U}^{k+1}) \\ & + \mu(\tilde{\mathbf{B}}^{k+1} - \frac{1}{\mu}\mathbf{M}_1^k) + \mu(\mathbf{H}^{k+1} - \frac{1}{\mu}\mathbf{M}_3^k){\mathbf{X}^k}^T, \end{aligned} \tag{22}$$

$$\begin{aligned} \mathbf{R}_6 &= (\tilde{\mathbf{A}}^{k+1})^T(\mathbf{H}^{k+1} - \frac{1}{\mu}\mathbf{M}_3^k) \\ & + (\mathbf{A}^{k+1})^T(\mathbf{J}^{k+1} - \frac{1}{\mu}\mathbf{M}_4^k) + {\mathbf{U}^{k+1}}^T - \frac{1}{\mu}{\mathbf{M}_5^k}^T. \end{aligned} \tag{23}$$

For detailed derivations of Algorithm 1, we refer the readers to the supplementary materials.

Convergence Analysis. We provide a partial result on the convergence of the proposed Algorithm 1 by virtue of KKT conditions of problem (9). To simplify notations, let us define

$$\Omega = (\mathbf{H}, \mathbf{B}, \mathbf{J}, \mathbf{U}, \tilde{\mathbf{B}}, \mathbf{A}, \tilde{\mathbf{A}}, \mathbf{X}, \mathbf{M}). \tag{24}$$

Theorem 4. Let $\{\Omega_k\}_{k=1}^\infty$ be a sequence generated by Algorithm 1 that satisfies the condition

$$\lim_{k\to\infty}(\Omega^{k+1} - \Omega^k) = 0. \tag{25}$$

Then any accumulation point of $\{\Omega^k\}_{k=1}^\infty$ is a KKT point of problem (15). Consequently, any accumulation point of $(\mathbf{A}^k, \tilde{\mathbf{A}}^k, \mathbf{X}^k)_k^\infty$ is a KKT point of problem (9).

Corollary 1. Whenever $\{\Omega_k\}_{k=1}^\infty$ converges, it converges to a KKT point.

As a consequence, for Algorithm 1, it is enough that we only need to choose a proper termination parameter $\epsilon > 0$, and use the termination condition:

$$\|\mathbf{Y} - \mathbf{A}^k\mathbf{X}^k\|_F^2 \leq \epsilon \ \text{ and } \ \|\mathbf{A}^{k+1} - \mathbf{A}^k|_F^2 \leq \epsilon. \tag{26}$$

All proofs of above theorem and corollary can be found in supplementary materials

6. Experiments and Discussions

Experiments were carried out on multiple real-world data sets. Throughout the experiments, we set ANMF parameters as $\alpha = 0.6$, $\beta = 10^{-5}$, $\gamma = 10^{-3}$, $\lambda = 10^{-3}$, and $\mu = 1$. For detailed description of datasets and more comprehensive results, we refer the readers to supplementary materials.

6.1. Comparison with Baselines

Some representative methods, including Standard Nonnegative Matrix Factorization (SNMF), $L_{2,1}$-norm based NMF model (Ding et al., 2006), Orthogonal Nonnegative Matrix Factorization (ONMF) (Kong et al., 2011), and Capped norm Nonnegative Matrix Factorization (CNMF) (Gao et al., 2015), are compared with the ANMF. It should noted that the main novelty of this paper is to consider potential test adversaries in modeling, not the robust characterization for noise. Thus, it is unfair to compare our method with some robust methods such as Correntropy induced NMF model and Truncated CauchyNMF model.

The detailed results for clustering accuracy and normalized mutual information results are shown in Table 2 and Table 3 (The best results are marked in bold). It can be observed that the advantage of ANMF is quite evident. Although $L_{2,1}$-norm based NMF is a robust NMF method, the ignoring of test adversaries leads to the undesired performance. Compared to NMF, ONMF achieves the better results, which indicates that the orthogonal constraint *w.r.t.* weight matrix **X** is helpful for improving the performance. However, our

Table 1. Description of Benchmark Datasets

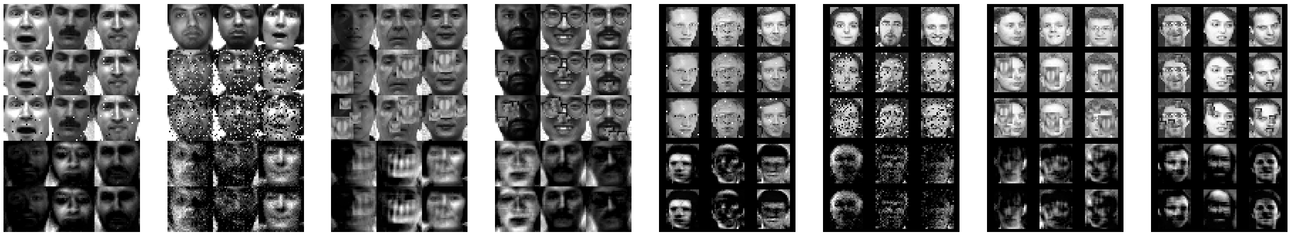
Dataset	Number of Instances	Dimensions	Classes	Category
MNIST	150	784	10	image
Yale	165	1024	15	image
ORL	400	644	40	image
UMIST	575	644	20	image
COIL-20	1440	1024	20	image
USPS	9298	256	10	image
BBCsports	737	4613	5	text
BBCNews	2225	9635	5	text
WebKB	4199	7770	4	text
Reuters	9298	256	10	text
RCV	9625	29992	4	text
TDT2	9394	36771	30	text

Table 2. ACC of noise-free Real Datasets. The best results are marked in bold.

Dataset	NMF	ONMF	$L2$, 1-norm NMF	CNMF	ANMF
MNIST	0.7933(±0.0497)	0.7987(±0.0441)	0.8027(±0.0410)	0.7947(±0.0228)	**0.8067**(±0.0490)
Yale	0.4388(±0.0233)	0.4145(±0.0360)	0.4424(±0.0235)	0.4036(±0.0380)	**0.4509**(±0.0164)
ORL	0.7005(±0.0060)	0.6420(±0.0356)	0.6895(±0.0139)	0.5935(±0.0323)	**0.7305**(±0.0294)
UMIST	0.4880(±0.0285)	0.4616(±0.0295)	0.4845(±0.0255)	0.4442(±0.0235)	**0.4946**(±0.0186)
COIL-20	0.6692(±0.0215)	0.6626(±0.0264)	0.6578(±0.0130)	0.6601(±0.0300)	**0.6833**(±0.0162)
USPS	0.7468(±0.0004)	0.7738(±0.0003)	0.7550(±0.0002)	0.7429(±0.0050)	**0.7780**(±0.0002)
BBCSport	0.9493(±0.0007)	0.9460(±0.0024)	0.9468(±0.0006)	0.9327(±0.0064)	**0.9531**(±0.0031)
BBC	0.9604(±0.0011)	0.9619(±0.0028)	0.9597(±0.0002)	0.9202(±0.0032)	**0.9649**(±0.0010)
WebKB	0.6619(±0.0095)	0.6657(±0.0038)	0.6618(±0.0083)	0.6525(±0.0117)	**0.6672**(±0.0084)
Reuters	0.7836(±0.0059)	0.7495(±0.0164)	0.7788(±0.0071)	0.7197(±0.0112)	**0.8047**(±0.0098)
RCV	0.6458(±0.0194)	0.6493(±0.0054)	0.6420(±0.0183)	0.6280(±0.0021)	**0.6516**(±0.0137)
TDT2	0.8546(±0.0067)	0.8246(±0.0119)	0.8448(±0.0046)	0.8062(±0.0150)	**0.8638**(±0.0176)

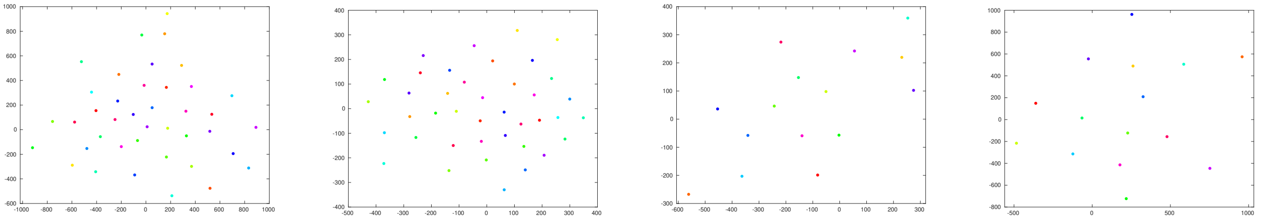
Table 3. ACC of noisy Real Datasets. The best results are marked in bold.

Noise	Dataset	NMF	ONMF	$L2$, 1-norm NMF	CNMF	ANMF
SP	MNIST	0.8067(±0.0464)	0.8093(±0.0379)	0.8080(±0.0477)	0.8067(±0.0254)	**0.8160**(±0.0421)
	Yale	0.3879(±0.0321)	0.3527(±0.0248)	0.3806(±0.0168)	0.3576(±0.0223)	**0.4036**(±0.0180)
	UMIST	0.4800(±0.0150)	0.4602(±0.0268)	0.4814(±0.0086)	0.4275(±0.0162)	**0.5078**(±0.0124)
	ORL	0.6155(±0.0252)	0.5475(±0.0083)	0.6225(±0.0275)	0.5350(±0.0173)	**0.6670**(±0.0248)
	COIL-20	0.6723(±0.0247)	0.6547(±0.0190)	0.6762(±0.0175)	0.6782(±0.0316)	**0.6830**(±0.0194)
	USPS	0.7542(±0.0004)	0.7716(±0.0003)	0.7592(±0.0003)	0.7505(±0.0058)	**0.7793**(±0.0002)
Pixel	MNIST	0.7880(±0.0417)	0.7733(±0.0194)	0.7800(±0.0481)	0.7453(±0.0202)	**0.8027**(±0.0293)
	Yale	0.3867(±0.0301)	0.3261(±0.0464)	0.3576(±0.0424)	0.3394(±0.0346)	**0.4012**(±0.0286)
	UMIST	0.4706(±0.0261)	0.4483(±0.0170)	0.4720(±0.0235)	0.4310(±0.0269)	**0.4866**(±0.0223)
	ORL	0.5370(±0.0141)	0.4850(±0.0157)	0.5145(±0.0192)	0.4650(±0.0190)	**0.5600**(±0.0366)
	COIL-20	0.6829(±0.0117)	0.6469(±0.0209)	0.6850(±0.0293)	0.6229(±0.0345)	**0.6924**(±0.0337)
	USPS	0.7520(±0.0002)	0.7638(±0.0009)	0.7527(±0.0007)	0.7269(±0.0003)	**0.7654**(±0.0006)
regular	MNIST	0.8107(±0.0494)	0.8040(±0.0376)	0.8093(±0.0543)	0.7920(±0.0311)	**0.8160**(±0.0423)
	Yale	0.3597(±0.0175)	0.3547(±0.0309)	0.3651(±0.0330)	0.3519(±0.0158)	**0.3852**(±0.0273)
	UMIST	0.4525(±0.0302)	0.4737(±0.0242)	0.4710(±0.0259)	0.4223(±0.0248)	**0.4828**(±0.0251)
	ORL	0.5465(±0.0243)	0.5145(±0.0288)	0.5520(±0.0198)	0.4695(±0.0368)	**0.5680**(±0.0207)
	COIL-20	0.5274(±0.0209)	0.5145(±0.0121)	0.5278(±0.0054)	0.5293(±0.0284)	**0.5315**(±0.0198)
	USPS	0.5210(±0.0005)	0.5296(±0.0074)	0.5195(±0.0018)	0.5306(±0.0078)	**0.5327**(±0.0065)
irregular	MNIST	0.2493(±0.0037)	0.2440(±0.0060)	0.2427(±0.0060)	0.2480(±0.0056)	**0.2497**(±0.0163)
	Yale	0.5468(±0.0261)	0.4982(±0.0458)	0.5406(±0.0266)	0.4861(±0.0262)	**0.5549**(±0.0301)
	UMIST	0.2247(±0.0056)	0.2115(±0.0089)	0.2235(±0.0087)	0.2136(±0.0051)	**0.2271**(±0.0057)
	ORL	0.3250(±0.0127)	0.2880(±0.0132)	0.3230(±0.0110)	0.2780(±0.0082)	**0.3485**(±0.0146)
	COIL-20	0.6792(±0.0202)	0.6706(±0.0228)	0.6782(±0.0181)	0.6586(±0.0177)	**0.6827**(±0.0145)
	USPS	0.7388(±0.0002)	**0.7602**(±0.0001)	0.7455(±0.0001)	0.7320(±0.0001)	0.7559(±0.0003)



(a) S & P (b) pixel (c) regular (d) irregular (e) S & P (f) pixel (g) regular (h) irregular

Figure 1. Illustrations of face datasets Yale (a)-(d) and ORL (e)-(h) with different types of noises (pixel, regular, irregular). From top row to bottom row: origin, noisy data, noisy $\mathbf{Z}$, $\mathbf{A}$, $\tilde{\mathbf{A}}$



(a) $\tilde{\mathbf{A}}$ on ORL (b) $\mathbf{A}$ on ORL (c) $\tilde{\mathbf{A}}$ on Yale (d) $\mathbf{A}$ on Yale

Figure 2. Visualizing Feature Matrices $\mathbf{A}$ and $\tilde{\mathbf{A}}$ via T-SNE on ORL and Yale Datasets

Table 4. ACC of Real Datasets with Salt & Pepper Noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
MNIST	0.7893(±0.0342)	0.8067(±0.0464)	0.8093(±0.0379)	0.8080(±0.0477)	0.8067(±0.0254)	**0.8160**(±0.0421)
Yale	0.3503(±0.0259)	0.3879(±0.0321)	0.3527(±0.0248)	0.3806(±0.0168)	0.3576(±0.0223)	**0.4036**(±0.0180)
UMIST	0.4734(±0.0157)	0.4800(±0.0150)	0.4602(±0.0268)	0.4814(±0.0086)	0.4275(±0.0162)	**0.5078**(±0.0124)
ORL	0.5720(±0.0195)	0.6155(±0.0252)	0.5475(±0.0083)	0.6225(±0.0275)	0.5350(±0.0173)	**0.6670**(±0.0248)
COIL-20	0.6678(±0.0123)	0.6723(±0.0247)	0.6547(±0.0190)	0.6762(±0.0175)	0.6782(±0.0316)	**0.6830**(±0.0194)
USPS	0.7706(±0.0008)	0.7542(±0.0004)	0.7716(±0.0003)	0.7592(±0.0003)	0.7505(±0.0058)	**0.7793**(±0.0002)

Table 5. NMI of Real Datasets with Salt & Pepper Noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
Yale	0.4008(±0.0268)	0.4393(±0.0261)	0.4070(±0.0277)	0.4255(±0.0229)	0.4144(±0.0203)	**0.4509**(±0.0168)
UMIST	0.5917(±0.0123)	0.5941(±0.0168)	0.5821(±0.0198)	0.5959(±0.0098)	0.5467(±0.0157)	**0.6123**(±0.0083)
ORL	0.7651(±0.0136)	0.7807(±0.0132)	0.7410(±0.0049)	0.7844(±0.0234)	0.7265(±0.0156)	**0.8077**(±0.0154)
MNIST	0.7464(±0.0175)	0.7648(±0.0241)	0.7689(±0.0217)	0.7676(±0.0242)	0.7580(±0.0179)	**0.7776**(±0.0252)
USPS	0.6565(±0.0011)	0.6349(±0.0003)	0.6581(±0.0002)	0.6478(±0.0001)	0.6356(±0.0053)	**0.6594**(±0.0006)
COIL-20	0.7528(±0.0149)	0.7506(±0.0174)	0.7497(±0.0131)	0.7546(±0.0079)	0.7512(±0.0136)	**0.7584**(±0.0173)

method is more competitive than others on the all databases. Therefore, considering the adversarial perturbations in modeling can increase the robustness of NMF.

Figure 1 provides the visual decomposition results on Yale and ORL under different noise. For each figure, we present the noise-free images, $\mathbf{Z}$, $\tilde{\mathbf{Z}}$, $\mathbf{A}$ and $\tilde{\mathbf{A}}$ from top to bottom. The results indicate that our method can successfully avoid redundancy of features and learn the desired feature matrices $\mathbf{A}$ and $\tilde{\mathbf{A}}$ with good representation performance. Figure 2 illustrates distribution of $\mathbf{A}$ and $\tilde{\mathbf{A}}$ with dimensionality reduced using T-SNE (Maaten & Hinton, 2008). We can see that all data points are fully separated.

To demonstrate the robustness of the proposed method, we also include the experimental results on image data corrupted by various noise. For each type we corrupt images successively to generate $\tilde{\mathbf{X}}$, $\mathbf{Z}$ and $\tilde{\mathbf{Z}}$ (which includes three

Table 6. ACC of Real Datasets with Corrupt Pixel Noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
Yale	0.3188(±0.0355)	0.3867(±0.0301)	0.3261(±0.0464)	0.3576(±0.0424)	0.3394(±0.0346)	**0.4012**(±0.0286)
UMIST	0.4557(±0.0134)	0.4706(±0.0261)	0.4483(±0.0170)	0.4720(±0.0235)	0.4310(±0.0269)	**0.4866**(±0.0223)
ORL	0.4775(±0.0137)	0.5370(±0.0141)	0.4850(±0.0157)	0.5145(±0.0192)	0.4650(±0.0190)	**0.5600**(±0.0366)
MNIST	0.7547(±0.0145)	0.7880(±0.0417)	0.7733(±0.0194)	0.7800(±0.0481)	0.7453(±0.0202)	**0.8027**(±0.0293)
USPS	0.7619(±0.0007)	0.7520(±0.0002)	0.7638(±0.0009)	0.7527(±0.0007)	0.7269(±0.0003)	**0.7654**(±0.0006)
COIL-20	0.6332(±0.0262)	0.6829(±0.0117)	0.6469(±0.0209)	0.6850(±0.0293)	0.6229(±0.0345)	**0.6924**(±0.0337)

Table 7. NMI of Real Datasets with Corrupt Pixel Noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
Yale	0.3618(±0.0334)	0.4200(±0.0274)	0.3755(±0.0466)	0.3983(±0.0360)	0.3908(±0.0287)	**0.4332**(±0.0226)
UMIST	0.5627(±0.0061)	0.5760(±0.0162)	0.5542(±0.0115)	0.5782(±0.0143)	0.5416(±0.0247)	**0.5857**(±0.0147)
ORL	0.6868(±0.0120)	0.7195(±0.0148)	0.6830(±0.0132)	0.7120(±0.0134)	0.6722(±0.0161)	**0.7295**(±0.0240)
MNIST	0.7170(±0.0278)	0.7576(±0.0306)	0.7540(±0.0202)	0.7534(±0.0364)	0.7192(±0.0180)	**0.7642**(±0.0144)
USPS	0.6487(±0.0007)	0.6358(±0.0002)	0.6486(±0.0008)	0.6358(±0.0005)	0.6079(±0.0007)	**0.6495**(±0.0011)
COIL-20	0.7548(±0.0139)	0.7670(±0.0100)	0.7529(±0.0151)	0.7659(±0.0174)	0.7369(±0.0181)	**0.7685**(±0.0093)

Table 8. ACC of Real Datasets with regular patch noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
MNIST	0.7947(±0.0296)	0.8107(±0.0494)	0.8040(±0.0376)	0.8093(±0.0543)	0.7920(±0.0311)	**0.8160**(±0.0423)
Yale	0.3464(±0.0308)	0.3597(±0.0175)	0.3547(±0.0309)	0.3651(±0.0330)	0.3519(±0.0158)	**0.3852**(±0.0273)
UMIST	0.4525(±0.0302)	0.4737(±0.0242)	0.4449(±0.0272)	0.4710(±0.0259)	0.4223(±0.0248)	**0.4828**(±0.0251)
ORL	0.5430(±0.0151)	0.5465(±0.0243)	0.5145(±0.0288)	0.5520(±0.0198)	0.4695(±0.0368)	**0.5680**(±0.0207)
COIL-20	0.5188(±0.0095)	0.5274(±0.0209)	0.5145(±0.0121)	0.5278(±0.0054)	0.5293(±0.0284)	**0.5315**(±0.0198)
USPS	0.5218(±0.0057)	0.5210(±0.0005)	0.5296(±0.0074)	0.5195(±0.0018)	0.5306(±0.0078)	**0.5327**(±0.0065)

Table 9. NMI of Real Datasets with regular patch Noise. The best results are marked in bold.

Dataset	K-Means	NMF	ONMF	$\ell_{2,1}$ NMF	CNMF	ANMF
MNIST	0.7538(±0.0141)	0.7731(±0.0237)	0.7725(±0.0197)	0.7723(±0.0327)	0.7420(±0.0222)	**0.7775**(±0.0217)
Yale	0.3805(±0.0467)	0.3967(±0.0184)	0.3860(±0.0175)	0.3989(±0.0536)	0.3905(±0.0267)	**0.4166**(±0.0319)
UMIST	0.5311(±0.0247)	0.5458(±0.0167)	0.5204(±0.0260)	0.5424(±0.0207)	0.4942(±0.0210)	**0.5511**(±0.0209)
ORL	0.7117(±0.0150)	0.7146(±0.0169)	0.6876(±0.0266)	0.7159(±0.0140)	0.6579(±0.0334)	**0.7257**(±0.0141)
COIL-20	0.6104(±0.0147)	0.6073(±0.0153)	0.6138(±0.0119)	0.6027(±0.0064)	**0.6157**(±0.0218)	**0.6157**(±0.0155)
USPS	0.3924(±0.0079)	0.3720(±0.0031)	0.3846(±0.0039)	0.3730(±0.0065)	0.3703(±0.0173)	**0.4001**(±0.0137)

stages of noise). In detail, two types of noise are considered: salt & pepper noise, in which we corrupt 10% pixels in each stage and the results are summarized in Table 4 and Table 5; random corrupted pixels, in which we corrupt 20%, 5%, and 5% pixels in each stage and the results are summarized in Table 6 and Table 7; random corrupted regular patches (), in which we corrupt 50%, 25%, and 25% pixels in each stage (see Figure 1) and the results are summarized in Table 8 and Table 9. The robustness of the proposed method is clearly verified according to these tables. Due to the space limitation, experimental results for *more types of noise*, *convergence curves of Algorithm 1* and *the comparison with other methods* can be found in supplementary materials.

7. Conclusion

This paper focuses on nonnegative matrix factorization problem. To provide the robustness against real perturbations, we propose a new Adversarial Nonnegative Matrix Factorization model The adversarial perturbations of **Y** is used to learn the desire feature matrix **A** and weight matrix **X**. Different from the traditional NMF models which only focus on the regular data points, our models emphasizes potential test adversaries that are beyond the pre-defined constraints.

We formulate the proposed model as a bilevel optimization problem and utilize ADMM to solve it with convergence guarantee. Experimental results on real data sets validate the effectiveness and robustness of the proposed algorithm.

Acknowledgements

This work was partially supported by U.S. NSF IIS 1836945, IIS 1836938, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956.

References

Bucak, S. S. and Gunsel, B. Video content representation by incremental non-negative matrix factorization. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pp. II–113. IEEE, 2007.

Cai, D., He, X., Han, J., and Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

Chartrand, R. and Wohlberg, B. A nonconvex admm algorithm for group sparsity with sparse groups. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6009–6013. IEEE, 2013.

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

Du, L., Li, X., and Shen, Y.-D. Robust nonnegative matrix factorization via half-quadratic minimization. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 201–210. IEEE, 2012.

Farnia, F., Zhang, J. M., and Tse, D. Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457*, 2018.

Fortin, M. and Glowinski, R. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*, volume 15. Elsevier, 2000.

Gabay, D. and Mercier, B. *A dual algorithm for the solution of non linear variational problems via finite element approximation*. Institut de recherche d'informatique et d'automatique, 1975.

Gao, H., Nie, F., Cai, W., and Huang, H. Robust capped norm nonnegative matrix factorization: Capped norm nmf. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 871–880. ACM, 2015.

Geng, B., Tao, D., Xu, C., Yang, L., and Hua, X.-S. Ensemble manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1227–1233, 2012.

Gonzalez, E. F. and Zhang, Y. Accelerating the lee-seung algorithm for nonnegative matrix factorization. Technical report, 2005.

Großhans, M., Sawade, C., Brückner, M., and Scheffer, T. Bayesian games for adversarial regression problems. In *International Conference on Machine Learning*, pp. 55–63, 2013.

Guan, N., Tao, D., Luo, Z., and Yuan, B. Nenmf: An optimal gradient method for nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 60(6): 2882–2898, 2012.

Guan, N., Liu, T., Zhang, Y., Tao, D., and Davis, L. S. Truncated cauchy non-negative matrix factorization for robust subspace learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Haeffele, B., Young, E., and Vidal, R. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International conference on machine learning*, pp. 2007–2015, 2014.

Hajinezhad, D., Chang, T.-H., Wang, X., Shi, Q., and Hong, M. Nonnegative matrix factorization using admm: Algorithm and convergence analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4742–4746. IEEE, 2016.

Huang, J., Nie, F., Huang, H., and Ding, C. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):11, 2014.

Kim, J., Monteiro, R. D., and Park, H. Group sparsity in nonnegative matrix factorization. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 851–862. SIAM, 2012.

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

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788, 1999.

Li, B. and Vorobeychik, Y. Feature cross-substitution in adversarial classification. In *Advances in neural information processing systems*, pp. 2087–2095, 2014.

Lin, C.-J. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

Liu, T., Gong, M., and Tao, D. Large-cone nonnegative matrix factorization. *IEEE transactions on neural networks and learning systems*, 28(9):2129–2142, 2016.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Mei, J.-J., Dong, Y., Huang, T.-Z., and Yin, W. Cauchy noise removal by nonconvex admm with convergence guarantees. *Journal of Scientific Computing*, 74(2):743–766, 2018.

Tong, L., Yu, S., Alfeld, S., and Vorobeychik, Y. Adversarial regression with multiple learners. *arXiv preprint arXiv:1806.02256*, 2018.

Wang, Y., Yin, W., and Zeng, J. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, pp. 1–35, 2015.

Zhi, R., Flierl, M., Ruan, Q., and Kleijn, W. B. Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):38–52, 2011.