Adversarially-Trained Nonnegative Matrix Factorization

Ting Cai, Vincent y F Tan, Cédric Févotte

Adversarially-Trained Nonnegative Matrix Factorization

Ting Cai, Vincent Y. F. Tan, *Senior Member, IEEE*, and Cédric Févotte, *Senior Member, IEEE*

arXiv:2104.04757v2 [cs.LG] 22 Jun 2021

Abstract—We consider an adversarially-trained version of the nonnegative matrix factorization, a popular latent dimensionality reduction technique. In our formulation, an attacker adds an arbitrary matrix of bounded norm to the given data matrix. We design efficient algorithms inspired by adversarial training to optimize for dictionary and coefficient matrices with enhanced generalization abilities. Extensive simulations on synthetic and benchmark datasets demonstrate the superior predictive performance on matrix completion tasks of our proposed method compared to state-of-the-art competitors, including other variants of adversarial nonnegative matrix factorization.

Index Terms—Adversarial Training, Non-negative Matrix Factorization, Matrix Completion,

I. INTRODUCTION

NONNEGATIVE matrix factorization (NMF) is a dictionary learning technique for learning non-subtractive and parts-based representations of given nonnegative data [1]. The mathematical formulation consists in decomposing a given nonnegative data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$ into two nonnegative matrices—the dictionary matrix $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and the coefficient matrix $\mathbf{H} \in \mathbb{R}_+^{K \times N}$ such that $\mathbf{V} \approx \mathbf{WH}$. To find an approximate decomposition, one solves the non-convex problem (known throughout the paper as *standard NMF*)

$$\min_{\mathbf{W}, \mathbf{H} \geq \mathbf{0}} \|\mathbf{V} - \mathbf{WH}\|_F^2, \tag{1}$$

where $\mathbf{A} \geq \mathbf{0}$ means all entries of the matrix $\mathbf{A}$ are nonnegative. Popularized by Lee and Seung [1], NMF has found a variety of applications in matrix completion [2], document clustering [3] and audio signal processing [4]. For a thorough review on NMF, the reader is referred to the book by Gillis [5].

As demonstrated in the landmark work by Goodfellow *et al.* [6], machine learning methods are vulnerable to adversarial attacks on the observed data. To mitigate this and to improve generalization, adversarial training (AT) has recently come to the fore. AT has been shown to improve the robustness of deep learning models in [7] and [8]. More recently, Sinha *et al.* [9] and Farnia *et al.* [10] showed theoretically that AT improves generalization in generic machine learning tasks.

In this paper, we seek to improve the predictive performance of NMF on matrix completion tasks by considering an adversarially-trained version of the standard NMF objective in (1). While there is a previous work by Luo *et al.* [11], called *adversarial NMF* or *ANMF*, which attempts to address

T. Cai and V. Y. F. Tan are with the National University of Singapore (emails: elect@nus.edu.sg, vtan@nus.edu.sg). C. Févotte is with IRIT, Université de Toulouse, CNRS, Toulouse, France (email: cedric.fevotte@irit.fr). This work is funded by a Singapore NRF Fellowship (R-263-000-D02-281) and by the European Research Council (ERC FACTORY-CoG-6681839).

a similar problem, here we highlight some salient differences between [11] and our formulation. Luo *et al.* [11] applied adversarial training *only* to the dictionary matrix $\mathbf{W}$. Their formulation necessitates the introduction of a so-called *instance-specific target* $\mathbf{Z}$ (a hyperparameter) for the adversary; this target $\mathbf{Z}$ is not easy to choose in a principled way in practice. The adversary learns a dictionary matrix $\tilde{\mathbf{W}}$ such that $\tilde{\mathbf{W}}\mathbf{H}$ is close to $\mathbf{Z}$. The learner then optimizes a linear combination of two objectives—one objective encouraging $\tilde{\mathbf{W}}\mathbf{H}$ to be close to the data matrix $\mathbf{V}$ and another encouraging $\mathbf{WH}$ to be close to $\mathbf{V}$. The authors used an ADMM algorithm for this bilevel optimization and demonstrated that ANMF is able to increase the robustness of NMF model on clustering tasks. We consider the case in which the attacker is allowed to attack $\mathbf{V}$ without being instance-specific, i.e., we consider the problem of minimizing $(\mathbf{W}, \mathbf{H}) \mapsto \max_{\mathbf{R}} \|\mathbf{V} + \mathbf{R} - \mathbf{WH}\|_F^2$ where $\mathbf{R}$ belongs to a bounded set $\mathcal{R}$. In contrast to ANMF, our formulation has only *one scalar* hyperparameter λ. We derive efficient updates for $\mathbf{R}$ and $(\mathbf{W}, \mathbf{H})$, which is known as *adversarially-trained NMF* or *AT-NMF*. We compare the prediction performance of AT-NMF to ANMF and standard NMF and show the superior performance on matrix completion tasks for three benchmark datasets commonly used in NMF.

II. ADVERSARIALLY-TRAINED NMF

We now introduce the formulation for AT-NMF. There exists an adversarial attacker who adds an *arbitrary* matrix $\mathbf{R} \in \mathbb{R}^{F \times N}$ to the data matrix $\mathbf{V}$. The *adversarial matrix* $\mathbf{R}$ maximizes the Frobenius norm between $\mathbf{V}$ and our desired factorization $\mathbf{WH}$; see the similar manner in which adversarial noise is added on to the training data in [9], [10]. Since the adversary typically has limited power, we assume that $\mathbf{R}$ belongs to a bounded set. Thus, AT-NMF can be formulated as

$$\min_{\mathbf{W}, \mathbf{H} \geq \mathbf{0}} \max_{\mathbf{R} \in \mathcal{R}} \|\mathbf{V} + \mathbf{R} - \mathbf{WH}\|_F^2 \tag{2}$$

where the constraint set $\mathcal{R} := \{\mathbf{R} : \|\mathbf{R}\|_F^2 \leq \epsilon, \mathbf{V} + \mathbf{R} \geq \mathbf{0}\}$. Here, $\epsilon > 0$ is a constant that indicates the adversary's power; the larger ϵ is, the larger the adversary's power and *vice versa*. We assume that $\mathbf{R}$ belongs to the set of Frobenius norm-bounded matrices and $\mathbf{V} + \mathbf{R}$ is mandated to be nonnegative so that for any fixed $\mathbf{R}$, the optimization in (2) corresponds to a standard NMF problem with effective data matrix $\mathbf{V} + \mathbf{R}$. We note that a similar min-max formulation has appeared in the recent literature [12] but the objective therein is different as the authors sought to only be robust to the β-divergence measure used and not perturbations to the data matrix $\mathbf{V}$.

The problem as stated in (2) is difficult to solve in general. Hence, we resort to a relaxation inspired by Lagrangian

duality [13]. We dualize the norm constraint $\|\mathbf{R}\|_{\mathrm{F}}^2 \leq \epsilon$ with a Lagrange multiplier $\lambda > 0$ so that the optimization problem becomes

$$\min_{\mathbf{W}, \mathbf{H} \geq \mathbf{0}} \max_{\mathbf{R}: \mathbf{V} + \mathbf{R} \geq \mathbf{0}} \|\mathbf{V} + \mathbf{R} - \mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2 - \lambda \|\mathbf{R}\|_{\mathrm{F}}^2. \quad (3)$$

Note that the regularization term has a *negative* coefficient because the inner optimization in (3) is a *maximization*. The interpretation of λ is dual to that of ϵ; it indicates the adversary's power. Indeed as $\lambda \to 0^+$, we see that the norm of $\mathbf{R}$ will be unbounded, indicating an extremely powerful adversary. Conversely, as $\lambda \to \infty$, the effect of the regularization term vanishes and the norm of $\mathbf{R}$ tends to zero, indicating that there is effectively no adversary.

The optimization problem in (3) can naturally be broken up into two parts, an *inner* maximization in which we optimize over $\mathbf{R}$ and an *outer* minimization in which we optimize over $(\mathbf{W}, \mathbf{H})$. The inner maximization can be equivalently rewritten as a minimization problem as

$$\mathbf{R}^* = \underset{\mathbf{R}: \mathbf{V} + \mathbf{R} \geq \mathbf{0}}{\arg\min} -\|\mathbf{V} + \mathbf{R} - \mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2 + \lambda \|\mathbf{R}\|_{\mathrm{F}}^2. \quad (4)$$

We show in the next section that the solution to this minimization problem is well-defined for $\lambda > 1$; indeed, it can be found in closed-form. After $\mathbf{R}^*$ has been found, one can then aim to minimize

$$\|\mathbf{V} + \mathbf{R}^* - \mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2 \quad (5)$$

over $\mathbf{W}, \mathbf{H} \geq \mathbf{0}$. By treating $\mathbf{V} + \mathbf{R}^*$ as a new data matrix, we can use Majorization-Minimization (MM) [14] to find approximate solutions for $\mathbf{W}$ and $\mathbf{H}$. We note that since we are using the Frobenius norm, there are alternatives to MM.

A. Update of $\mathbf{R}$ in (4)

We now provide details on how to solve for $\mathbf{R}^*$ in (4). Here the matrices $\mathbf{W}$ and $\mathbf{H}$ are fixed and we denote their product as $\hat{\mathbf{V}} = \mathbf{W}\mathbf{H}$. Thus, we can denote the objective as

$$g(\mathbf{R}) := -\|\mathbf{V} + \mathbf{R} - \hat{\mathbf{V}}\|_{\mathrm{F}}^2 + \lambda \|\mathbf{R}\|_{\mathrm{F}}^2, \quad (6)$$

where $\mathbf{R}$ is constrained such that $\mathbf{V} + \mathbf{R} \geq \mathbf{0}$. We note that this objective, by virtue of being the sum of two squared Frobenius norms, is *separable*, i.e., it decomposes into the sum of FN independent terms, i.e.,

$$g(\mathbf{R}) = \sum_{f=1}^{F} \sum_{n=1}^{N} \left[-(v_{fn} + r_{fn} - \hat{v}_{fn})^2 + \lambda r_{fn}^2 \right] \quad (7)$$

Thus, to minimize $g(\mathbf{R})$ over $\mathbf{R}$, it suffices to minimize each of the constituent terms in the sum over r_{fn}. In the following, we abbreviate v_{fn}, r_{fn} and $\hat{v}_{fn}$ to v, r and $\hat{v}$ respectively.

The problem thus reduces to the scalar optimization

$$\min_{r: v + r \geq 0} -(v + r - \hat{v})^2 + \lambda r^2 \quad (8)$$

which is equivalent to

$$\min_{r: v + r \geq 0} (\lambda - 1) r^2 - 2r(v - \hat{v}). \quad (9)$$

If $0 \leq \lambda < 1$, the objective function is strictly concave in r. Thus the optimal value of r is ∞, which indicates that

the adversary has unbounded power, an unrealistic scenario. When $\lambda = 1$, we again have undesirable solutions. Indeed, the objective function in (9) is linear. If $\hat{v} - v > 0$, then the solution is $r^* = -v$; if $\hat{v} = v$, regardless of the value of r, the function is 0; finally, if $\hat{v} - v < 0$, the solution is $r^* = \infty$. To avoid handling these cases separately, we avoid using $\lambda = 1$.

Thus the only meaningful case, and the case we consider in our experiments, is $\lambda > 1$. In this case, the objective function in (9) is strictly convex and it is easy to show by direct differentiation that the optimal constrained solution to (6) under the condition that $\mathbf{V} + \mathbf{R} \geq \mathbf{0}$ is

$$\mathbf{R} = \max \left\{ \frac{\mathbf{V} - \hat{\mathbf{V}}}{\lambda - 1}, -\mathbf{V} \right\}, \quad (10)$$

where the maximization operator is applied elementwise.

B. Update of $\mathbf{W}$ and $\mathbf{H}$ in (5)

Upon the optimization of $\mathbf{R}$ via the formula in (10), the optimization over $(\mathbf{W}, \mathbf{H})$ is standard if we regard $\mathbf{U} := \mathbf{V} + \mathbf{R}^*$ as the *effective data matrix*. The classical MM steps [2], [14] lead to multiplicative updates for $\mathbf{W}$ and $\mathbf{H}$ as follows

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^\top \mathbf{U}}{\mathbf{W}^\top \mathbf{W}\mathbf{H}} \quad \text{and} \quad \mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\mathbf{U}\mathbf{H}^\top}{\mathbf{W}\mathbf{H}\mathbf{H}^\top}, \quad (11)$$

where $\cdot$ and $\cdot / \cdot$ respectively denote elementwise multiplication and elementwise division respectively.

Just to recapitulate, we first update $\mathbf{R}$ via (10) and then update $(\mathbf{W}, \mathbf{H})$ via (11). These steps are shown in Algorithm 1.

C. Initialization of $(\mathbf{W}, \mathbf{H})$

We now describe how we initialize the dictionary and coefficient matrices $\mathbf{W}_{\mathrm{init}}$ and $\mathbf{H}_{\mathrm{init}}$. First, we sample each entry of each matrix independently from a Half-Normal distribution [15] with precision parameter $\gamma = 1$. Then, starting with these matrices, we run 5 standard MM steps according to (11) on the given data matrix $\mathbf{V}$ to obtain $\mathbf{W}_{\mathrm{init}}$ and $\mathbf{H}_{\mathrm{init}}$. The reason why we use this initialization method is to get a pair $(\mathbf{W}_{\mathrm{init}}, \mathbf{H}_{\mathrm{init}})$ such that $\hat{\mathbf{V}}_{\mathrm{init}} = \mathbf{W}_{\mathrm{init}} \mathbf{H}_{\mathrm{init}} \approx \mathbf{V}$. Otherwise, as (10) shows, $\mathbf{R}^*$ may be set to $-\mathbf{V}$ so that $\mathbf{U} = \mathbf{V} + \mathbf{R}^* = \mathbf{0}$, which renders the subsequent steps vacuous. This initialization is computationally cheap and effective.

D. Stopping Criteria of AT-NMF

Since $\mathbf{R}$ can be found in closed-form if $\lambda > 1$, no termination criterion for updating $\mathbf{R}$ is required. However, the optimization of $(\mathbf{W}, \mathbf{H})$ is found via iterative optimization according to (11). So, in this subsection, we discuss the criterion we employ to terminate this inner optimization before reverting to computing a new $\mathbf{R}$. We consider the *relative error* of successive iterates and terminate once this error falls below a certain threshold $\varepsilon_{\mathrm{in}}$, i.e., if $(\mathbf{W}^{(o,i)}, \mathbf{H}^{(o,i)})$ denotes the iterate of $(\mathbf{W}, \mathbf{H})$ at the oth outer iteration and i^{th} inner iteration and $\hat{\mathbf{V}}^{(o,i)} := \mathbf{W}^{(o,i)} \mathbf{H}^{(o,i)}$, we terminate the inner optimization once the inner iteration index i satisfies

$$\left\| \frac{\hat{\mathbf{V}}^{(o,i+1)} - \hat{\mathbf{V}}^{(o,i)}}{\hat{\mathbf{V}}^{(o,i)}} \right\|_{\mathrm{F}} < \varepsilon_{\mathrm{in}}. \quad (12)$$

for some $\varepsilon_{\text{in}} > 0$. The entire optimization is terminated once the outer iteration index o satisfies

$$\left\| \frac{\hat{\mathbf{V}}^{(o+1,i)} - \hat{\mathbf{V}}^{(o,i)}}{\hat{\mathbf{V}}^{(o,i)}} \right\|_{\mathrm{F}} < \varepsilon_{\text{out}} \qquad (13)$$

for some $\varepsilon_{\text{out}} > 0$ where i is the index of the inner iteration for which we terminated previously. In practice, we also terminate the inner and outer loops once preset number of iterations `maxInner` and `maxOuter` respectively are reached.

Algorithm 1: AT-NMF($\mathbf{V}, \lambda, K$)

1: **Input:** Data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$; Regularization param. λ; Stopping params. ε_{in} and ε_{out}; Common dim. K; Integers `maxInner` and `maxOuter`;
2: **Initialize:** $(\mathbf{W}, \mathbf{H})$ to $(\mathbf{W}_{\text{init}}, \mathbf{H}_{\text{init}})$ per Sec. II-C.
3: `outerNotConv` = TRUE;
4: **while** `outerNotConv` **do**
5: 　　Update $\mathbf{R}$ using (10);
6: 　　`innerNotConv` = TRUE
7: 　　**while** `innerNotConv` **do**
8: 　　　　Update $(\mathbf{W}, \mathbf{H})$ using (11);
9: 　　　　**if** Cond. (12) holds or `maxInner` reached **then**
10: 　　　　　　`innerNotConv` = FALSE
11: 　　　　**end if**
12: 　　**end while**
13: 　　**if** Cond. (13) holds or `maxOuter` reached **then**
14: 　　　　`outerNotConv` = FALSE
15: 　　**end if**
16: **end while**

III. NUMERICAL EXPERIMENTS

We perform numerical experiments on synthetic and real datasets to demonstrate the prediction and generalization efficacy of AT-NMF vis-à-vis standard NMF [1] and ANMF [11]. The hyperparameters that we use are `maxInner` = 1000, `maxOuter` = 100, $\varepsilon_{\text{in}} = \varepsilon_{\text{out}} = 0.01$. Our experiments all involve predicting entries of $\mathbf{V}$ that are held out. We denote the fraction of held-out entries as $\alpha \in (0, 1)$ which takes on values in $\{0.1, 0.2, \ldots, 0.8, 0.9\}$. That is, for $\alpha = 0.9$, we randomly remove 90% of the entries of $\mathbf{V}$, train the model (on standard NMF, ANMF and AT-NMF) on the remaining 10% of available entries and assess the prediction performance thereafter. Like standard NMF, Algorithm 1 can straightforwardly be adapted to handle missing values by applying a binary mask to $\mathbf{V}$, see, e.g., [2]. Note that r_{fn} is only computed for observed values. To be fair to all competing algorithms, we use the same stopping criterion as AT-NMF for NMF and the default parameters for ANMF [11]. The code to reproduce our experiments can be found at https://github.com/caiting123321/AT_NMF.

Let $\Gamma \subset \{1, \ldots, F\} \times \{1, \ldots, N\}$ be the set of held-out entries of $\mathbf{V}$ so $|\Gamma|/(FN) = \alpha$. We denote the prediction of $\{v_{fn} : (f, n) \in \Gamma\}$ as $\{\hat{v}_{fn} : (f, n) \in \Gamma\}$ where $\hat{v}_{fn} = [\mathbf{WH}]_{fn}$. Our performance metric is the *root mean-squared error* (RMSE), defined as

$$\text{RMSE} := \sqrt{\frac{1}{|\Gamma|} \sum_{(f,n) \in \Gamma} (v_{fn} - \hat{v}_{fn})^2}. \qquad (14)$$

TABLE I: RMSE of the synthetic dataset (Numbers in the brackets indicate the λ used in AT-NMF)

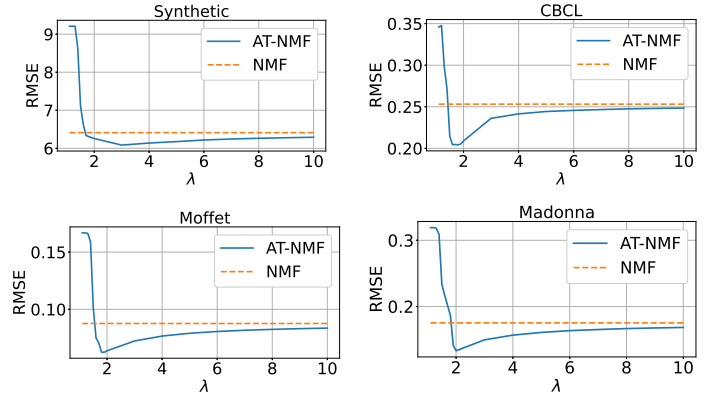| α	NMF	ANMF	AT-NMF (2)	AT-NMF (3)	AT-NMF (5)
0.3	5.37 ± 0.02	6.78 ± 0.17	5.41 ± 0.12	$\mathbf{5.11 \pm 0.03}$	5.20 ± 0.02
0.4	5.62 ± 0.03	6.92 ± 0.17	5.54 ± 0.08	$\mathbf{5.32 \pm 0.09}$	5.42 ± 0.04
0.5	6.41 ± 0.01	7.44 ± 0.09	6.27 ± 0.11	$\mathbf{6.05 \pm 0.03}$	6.18 ± 0.02
0.6	6.74 ± 0.02	7.61 ± 0.09	6.47 ± 0.07	$\mathbf{6.39 \pm 0.03}$	6.53 ± 0.02
0.7	7.30 ± 0.01	7.99 ± 0.06	7.02 ± 0.04	$\mathbf{6.94 \pm 0.01}$	7.10 ± 0.02
0.8	7.87 ± 0.01	8.30 ± 0.06	7.69 ± 0.04	$\mathbf{7.61 \pm 0.03}$	7.71 ± 0.00
0.9	8.45 ± 0.01	8.58 ± 0.06	8.44 ± 0.02	$\mathbf{8.34 \pm 0.02}$	8.35 ± 0.02

Fig. 1: Plots of RMSE against λ on the datasets when $\alpha = 0.5$

The smaller the RMSE, the better the prediction ability. All our results are averaged over 10 independent initializations (described in Sec. II-C). In Tables I–IV, we report the standard deviations in addition to the means across the 10 runs.

A. Simulations on a Synthetic Dataset

We describe experiments on synthetic data generated according to the model in [16]. We sampled $K = 5$ precision parameters γ_k according to the inverse-Gamma distribution with parameters $(a = 50, b = 70)$. Conditioned on these precisions, we sampled the elements of the kth column of $\mathbf{W} \in \mathbb{R}_+^{100 \times 5}$ and the kth row of $\mathbf{H} \in \mathbb{R}_+^{5 \times 50}$ independently from the Half-Normal distribution with precision parameters γ_k and $\mathbf{V} = \mathbf{WH}$. Our results are displayed in Table I.

First, we observe that all algorithms perform better when α is small. This is intuitive since there are more data to train on. More interestingly, AT-NMF outperforms standard NMF and ANMF consistently over different α for the values of λ chosen. For this dataset, ANMF performs worse than NMF. We believe that this because its hyperparameters, including the instance-specific target, have not been tuned (we are using the default ones). For this experiment, AT-NMF with $\lambda = 3$ outperforms AT-NMF with $\lambda = 2$ and $\lambda = 5$. Our final observation for the synthetic dataset pertains to scenario in which λ becomes large. In this case, the adversary's power is diminished; hence, AT-NMF reverts to standard NMF. Indeed, the performance of AT-NMF for large λ is close to that of standard NMF. See the top left plot of Fig. 1 where the RMSE of AT-NMF converges to that of NMF as $\lambda \to \infty$. This behavior was observed for the other real datasets as can be seen from the other subplots in Fig. 1. From our experience, the best choice of λ depends on the average magnitude of the entries of $\mathbf{V}$ and may require tuning to obtain the best performance.

TABLE II: RMSE of the CBCL dataset

α	NMF	ANMF	AT-NMF (2)	AT-NMF (5)
0.3	0.204 ± 0.001	0.189 ± 0.000	$\mathbf{0.176 \pm 0.001}$	0.195 ± 0.001
0.4	0.232 ± 0.001	0.217 ± 0.000	$\mathbf{0.192 \pm 0.001}$	0.224 ± 0.001
0.5	0.253 ± 0.001	0.241 ± 0.000	$\mathbf{0.209 \pm 0.001}$	0.244 ± 0.001
0.6	0.272 ± 0.000	0.262 ± 0.000	$\mathbf{0.227 \pm 0.000}$	0.262 ± 0.000
0.7	0.291 ± 0.000	0.285 ± 0.000	$\mathbf{0.248 \pm 0.001}$	0.283 ± 0.000
0.8	0.309 ± 0.000	0.306 ± 0.000	$\mathbf{0.277 \pm 0.001}$	0.303 ± 0.000
0.9	0.328 ± 0.000	0.326 ± 0.000	$\mathbf{0.314 \pm 0.000}$	0.324 ± 0.000

TABLE III: RMSE of the Moffet Dataset

α	NMF	ANMF	AT-NMF (2)	AT-NMF (5)
0.3	0.056 ± 0.000	0.050 ± 0.000	$\mathbf{0.041 \pm 0.001}$	0.049 ± 0.000
0.4	0.072 ± 0.000	0.067 ± 0.000	$\mathbf{0.052 \pm 0.001}$	0.064 ± 0.000
0.5	0.088 ± 0.000	0.084 ± 0.000	$\mathbf{0.064 \pm 0.000}$	0.079 ± 0.000
0.6	0.103 ± 0.000	0.100 ± 0.000	$\mathbf{0.079 \pm 0.000}$	0.095 ± 0.000
0.7	0.120 ± 0.000	0.117 ± 0.000	$\mathbf{0.097 \pm 0.000}$	0.113 ± 0.000
0.8	0.136 ± 0.000	0.134 ± 0.000	$\mathbf{0.117 \pm 0.000}$	0.131 ± 0.000
0.9	0.151 ± 0.000	0.150 ± 0.000	$\mathbf{0.140 \pm 0.000}$	0.148 ± 0.000
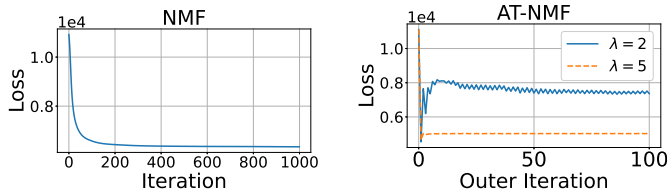
Fig. 2: Training losses of the CBCL dataset when $\alpha = 0.5$

TABLE IV: RMSE of the Madonna dataset

α	NMF	ANMF	AT-NMF (2)	AT-NMF (5)
0.3	0.120 ± 0.000	0.106 ± 0.000	$\mathbf{0.103 \pm 0.006}$	0.109 ± 0.000
0.4	0.147 ± 0.000	0.139 ± 0.000	$\mathbf{0.117 \pm 0.004}$	0.134 ± 0.000
0.5	0.175 ± 0.000	0.170 ± 0.000	$\mathbf{0.131 \pm 0.006}$	0.161 ± 0.000
0.6	0.204 ± 0.000	0.201 ± 0.000	$\mathbf{0.158 \pm 0.003}$	0.190 ± 0.000
0.7	0.232 ± 0.000	0.229 ± 0.000	$\mathbf{0.185 \pm 0.005}$	0.219 ± 0.000
0.8	0.261 ± 0.000	0.260 ± 0.000	$\mathbf{0.227 \pm 0.002}$	0.251 ± 0.000
0.9	0.290 ± 0.000	0.289 ± 0.000	$\mathbf{0.271 \pm 0.000}$	0.284 ± 0.000
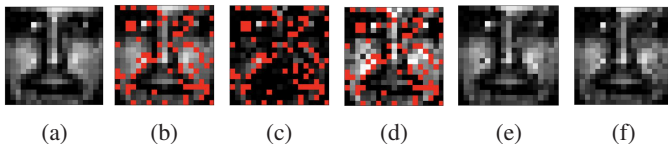
Fig. 3: (a) Original image $\mathbf{V}$; (b) Masked training image (missing pixels in red) $\mathbf{V}$; (c) Adversary's added-on masked image $\mathbf{R}^*$; (d) Adversarially trained masked image $\mathbf{V} + \mathbf{R}^*$; (e) Restored image using AT-NMF; (f) Restored image using NMF when $\alpha = 0.2$ and $\lambda = 2$.
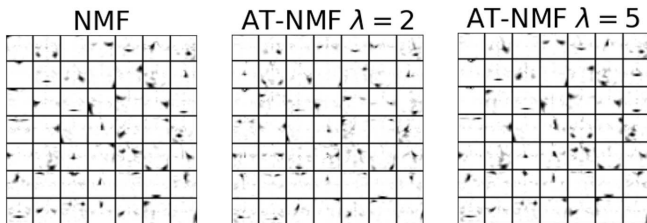
Fig. 4: Parts learnt on CBCL dataset when $\alpha = 0.1$

B. Simulations on the CBCL Dataset

We report experiments on the CBCL dataset [1]. This dataset consists of $N = 2429$ frontal-view facial images, each having $F = 19 \times 19 = 361$ pixels. Thus, $\mathbf{V} \in \mathbb{R}_+^{361 \times 2429}$. We normalized each image using the procedure outlined in [1]. Table II displays the results. We observe that AT-NMF with $\lambda = 2$ outperforms all competing algorithms. The top right plot in Fig. 1 also indicates that the performance of AT-NMF converges to that of NMF as $\lambda \to \infty$.

Fig. 2 shows the evolution of the objective function of standard NMF and AT-NMF with $\lambda = 2$ and $\lambda = 5$ for the *outer* iterations of Algorithm 1. For NMF, as expected, the objective function values converge. For AT-NMF, we observe that for small λ, the iterates exhibit long-term oscillatory behavior. This is because the MM property that guarantees that the loss is monotonically non-increasing is not present in AT-NMF where we maximize over $\mathbf{R}$ and minimize over $(\mathbf{W}, \mathbf{H})$. We leave the convergence of AT-NMF to future work. Notice

that despite reaching a higher objective value than NMF, the prediction performance is better. This is due to the built-in generalization ability of adversarial training in AT-NMF.

Fig. 3 shows an example of a masked training image and the corresponding image restored using AT-NMF. In AT-NMF, adding $\mathbf{R}^*$ on $\mathbf{V}$ intensifies the differences between the pixels, which results in a "training" image $\mathbf{V} + \mathbf{R}^*$ (subfigure (d)) whose features (eyes, nose, lower cheeks) become more distinctive. We observe that the restored image by AT-NMF (subfigure (e)) is similar to the original image (subfigure (a)).

Fig. 4 shows the parts learned. These are the columns of the basis matrix $\mathbf{W}$. We observe that AT-NMF retains the ability to extract parts-based representations. Specifically, the parts learned by AT-NMF with $\lambda = 2$ contain more noise in each basis image. This is natural due to the effect of adding $\mathbf{R}^*$.

C. Simulations on Hyperspectral Datasets

Finally, we describe experiments on two hyperspectral datasets, namely the Moffet [17] and Madonna [18] datasets, which are normalized in the same way as in [19]. This leads to $F = 165$ and $F = 160$ spectral bands in the Moffet and Madonna datasets respectively. The size of the images (number of pixels) is 50×50 so $N = 2500$. We set $K = 5$.

Tables III and IV display the results for these two datasets. We observe that AT-NMF with $\lambda = 2$ outperform all other algorithms for all α. The bottom plots in Fig. 1 also indicate that as $\lambda \to \infty$, AT-NMF reverts to standard NMF.

IV. CONCLUSION

We considered a formulation of AT-NMF in which an adversary adds a matrix $\mathbf{R}$ of bounded norm to the data matrix $\mathbf{V}$. We designed an efficient algorithm to solve the min-max problem. This algorithm "shapes" $\mathbf{R}$ in a way such that the Frobenius norm of $\mathbf{V} + \mathbf{R}$ and the approximating matrix $\mathbf{WH}$ is maximized. We show, through extensive numerical experiments, that AT-NMF outperforms standard NMF and ANMF [11] in terms of prediction. Designing algorithms for other divergence measures such as β-divergences [4], different bounded sets $\mathcal{R}$ and online implementations of NMF [20] would be interesting. Quantifying the generalization ability is also an avenue for future theoretical research.

REFERENCES

[1] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[2] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, 2011.

[3] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proc. of the 26th International ACM SIGIR Conference*, pages 267–273, 2003.

[4] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[5] Nicolas Gillis. *Nonnegative Matrix Factorization*. Society for Industrial & Applied Mathematics, 2020.

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of the 3rd International Conference on Learning Representations*, 2015.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of the 6th International Conference on Learning Representations*, 2018.

[8] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *Proc. of the 6th International Conference on Learning Representations*, 2018.

[9] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. In *Proc. of the 6th International Conference on Learning Representations*, 2018.

[10] Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *Proc. of the 7th International Conference on Learning Representations*, 2019.

[11] Lei Luo, Yanfu Zhang, and Heng Huang. Adversarial nonnegative matrix factorization. In *Proc. of the 37th International Conference on Machine Learning*, volume 119, pages 6479–6488, Jul 2020.

[12] Nicolas Gillis, Le Thi Khanh Hien, Valentin Leplat, and Vincent Y. F. Tan. Distributionally robust and multi-objective nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[14] David R. Hunter and Kenneth Lange. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000.

[15] F. C. Leone, L. S. Nelson, and R. B. Nottingham. The folded normal distribution. *Technometrics*, 4(3):543–550, 1961.

[16] Vincent Y. F. Tan and Cédric Févotte. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2013.

[17] Jet Propulsion Lab. (JPL). Aviris free data. California Inst. Technol., Pasadena, CA, 2006. [Online]. Available: http://aviris.jpl.nasa.gov/html/aviris.freedata.html.

[18] David Sheeren, Mathieu Fauvel, Sylvie Ladet, Anne Jacquin, Georges Bertoni, and Annick Gibon. Mapping ash tree colonization in an agricultural mountain landscape: Investigating the potential of hyperspectral imagery. In *Proc. IEEE Int. Conf. Geosci. Remote Sens. (IGARSS), Vancouver, Canada*, pages 3672–3675, July 2011.

[19] Cédric Févotte and Nicolas Dobigeon. Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 24(12):4810–4819, Dec 2015.

[20] Renbo Zhao and Vincent Y. F. Tan. Online nonnegative matrix factorization with outliers. *IEEE Transactions on Signal Processing*, 65(3):555–570, 2017.