Adversarial Tensor Factorization for Context-aware Recommendation

Huiyuan Chen
Electrical Engineering and Computer Science
Case Western Reserve University
hxc501@case.edu

Jing Li
Electrical Engineering and Computer Science
Case Western Reserve University
jingli@cwru.edu

ABSTRACT

Contextual factors such as time, location, or tag, can affect user preferences for a particular item. Context-aware recommendations are thus critical to improve both quality and explainability of recommender systems, compared to traditional recommendations that are solely based on user-item interactions. Tensor factorization machines have achieved the state-of-the-art performance due to their capability of integrating users, items, and contextual factors in one unify way. However, few work has focused on the robustness of a context-aware recommender system. Improving the robustness of a tensor-based model is challenging due to the sparsity of the observed tensor and the multi-linear nature of tensor factorization. In this paper, we propose *ATF*, a model that combines tensor factorization and adversarial learning for context-aware recommendations. Doing so allows us to reap the benefits of tensor factorization, while enhancing the robustness of a recommender model, and thus improves its eventual performance. Empirical studies on two real-world datasets show that the proposed method outperforms standard tensor-based methods.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Computing methodologies → Adversarial learning**.

KEYWORDS

Context-ware Recommendation; Tensor Factorization; Adversarial Learning; Deep Learning; Collaborative Filtering

ACM Reference Format:
Huiyuan Chen and Jing Li. 2019. Adversarial Tensor Factorization for Context-aware Recommendation. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3346987

1 INTRODUCTION

Beyond the user-item interactions, the *context* can provide abundant information about users' interests regarding items from different perspectives in many recommender systems [1]. For example, the context can be the time when a user purchases clothing on *Amazon* [8, 21], or the tag that a user annotates to a song on *Last.fm* [16, 18]. It has been shown that exploiting additional contextual factors can provide more accurate explainability and more relevant recommendations, compared to traditional recommender systems that are solely based on user-item interactions [1].

Factorization machines have recently become one of the preferred latent factor models for context-aware recommendations [11]. In particular, tensor-based models have achieved the state-of-the-art performance since they have the ability to integrate contextual information by learning a user-item-context tensor instead of the traditional user-item matrix [1, 3, 10, 17, 18, 20]. These methods in general attempt to recover the original triple-wise tensor interactions from several low dimensional latent factors [4, 11]. However, few work has focused on the robustness of a context-aware recommender system, which may fail to capture fine-grained and stable results due to noise data [12]. The main reason is that most current methods assume the multi-linear interaction between the latent factors, *i.e.*, the inner product between two latent factors in matrix factorization, or the outer product in tensor factorization. Small random perturbations on the parameters of linear models can lead to large *backward errors*, which has been discussed in matrix analysis [5] and recommender systems [9].

In addition to random perturbations, recent developments on adversarial learning [6, 7, 13, 19] have shown that many advanced classifiers are actually very vulnerable to adversarial perturbations, which are intentionally constructed to fool the classifiers in the training stage. Defensing against those adversarial perturbations are very useful to improve the vulnerabilities since models' equilibrium performance indicates their eventual success or failure. For example, in computer vision, adding small adversarial perturbations to an image can successfully cheat a well-trained classifier, albeit being quasi-imperceptible to the human eye [7]. In the field of information retrieval, generating difficult training data instances in an adversarial way can significantly improve both the robustness and the performance of the models [19]. Analogously, modeling the impacts of perturbations is the key to obtaining optimal performance for tensor-based models in context-aware recommendations.

Here we address the problem of adversarial perturbations on tensor-based models by developing Adversarial Tensor Factorization (ATF). Building upon adversarial learning techniques [7, 9, 13], our approach injects adversarial perturbations to tensor-based model parameters in context-aware recommendations. Intuitively, the adversarial perturbations tend to attack model parameters, while the model parameters aim to defense against those perturbations for self-improvement. We formulate a unified objective function to take both adversarial perturbations and model parameters into account. As such, our method reaps the benefits of tensor

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3346987

factorization, while enhancing the robustness of a recommender model, and thus improves its eventual performance. Empirical studies on two real-world datasets show that the proposed method substantially outperforms baselines.

2 RELATED WORK

Context-aware recommendations aim to analyze users' behaviors by considering contextual factors that are available from distinct sources [1]. Context can be, for instance, the time or location, users' social networks or user/item metadata [2, 10, 14, 17, 18, 20]. Integrating those contextual factors can help to improve recommender models. For example, MR [10] and Vista [8] were proposed to learn the time-evolving patterns by taking time into account. Such temporal factors show that users gradually pay/loss attention in different items as time goes by. In tag-aware recommendations, PITF [16] and RTF [14] studied the impacts of tags, which allowed users to describe an item with a list of words. In artistic recommendations, incorporating the items' metadata, such as their visual signals, with user-item interactions have achieved substantially more accurate performance than methods that solely make use of user-item rating matrices [8, 21]. Most current methods are built upon factorization machines [11]. However, few work focuses on the robustness of context-aware recommender systems. Several regularization techniques have been proposed to improve the robustness of factorization machines by adding various constraints on latent factors, such as L_1/L_2 norm, non-negativity and sparsity conditions [11, 12, 17, 21]. Nevertheless, they still suffer from small perturbations due to their intrinsic multi-linear structures, especially for adversarial perturbations [5, 9].

Motivated by the fast growing adversarial learning [6, 7, 13], researchers have attempted to generate adversarial noise to attack their machine learning models and then improve models' vulnerabilities, especially in the filed of computer vision and deep learning [7, 9, 13, 19]. Many of them are GAN-based frameworks [6], which contain a generative model and a discriminative model. The discriminative model aims to train a model from labelled and unlabelled data, while the generative model tries to attack the discriminative model by generating difficult samples in an adversarial way. Our ATF methodology is slightly different from those GAN-based methods, but is more relevant to recent developments [9, 13], in which the adversarial perturbations are directly injected to the model parameters. Such carefully crafted perturbations and model parameters can be estimated by solving an optimization problem.

3 METHODOLOGY

3.1 Problem Description

We use the notation in tag recommender [14, 16, 18], in which the contextual factor is the *tag*. Let $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{T}$ denote the set of users, items, and tags, respectively. The historical user-item-tag events can be represented by the set $\mathcal{D}$, *i.e.*, $(p, q, r) \in \mathcal{D}$ means that user p has tagged an item q with the tag r. We can use a tensor $\mathcal{X} \in \mathbb{R}^{P \times Q \times R}$ to indicate the interactions among the users, items and tags, where P, Q, R are the number of users, items and tags, respectively. if $(p, q, r) \in \mathcal{D}$, $\mathcal{X}_{pqr} = 1$, otherwise, $\mathcal{X}_{pqr} = 0$. In tag-aware recommendations, for a given user-item pair (p, q), the goal is to provide a list of tags that user p is likely to label item q. This can be achieved by predicting the missing entries (e.g., zero elements) in the tensor $\mathcal{X}$.

Although we focus on the tag-aware recommendations in this study, our method can be easily applied to other context-aware recommender systems, such as learning a $user \times item \times time$ tensor in time-aware recommendations [10].

3.2 Tensor Factorization

The Pairwise Interaction Tensor Factorization (PITF) has achieved better performance in comparison to tensor Tucker and CANDECOMP/PARAFAC (CP) models in context-aware recommendations [16]. Specifically, the PITF model decomposes tensor $\mathcal{X}$ via three pairs of inner products among the latent vectors of the user $(\mathbf{u}_p^{(v)}, \mathbf{u}_p^{(t)})$, item $(\mathbf{v}_q^{(u)}, \mathbf{v}_q^{(t)})$ and tag $(\mathbf{t}_r^{(u)}, \mathbf{t}_r^{(v)})$:

$$\hat{\mathcal{X}}_{pqr}(\Theta) = \langle \mathbf{u}_p^{(v)}, \mathbf{v}_q^{(u)} \rangle + \langle \mathbf{u}_p^{(t)}, \mathbf{t}_r^{(u)} \rangle + \langle \mathbf{v}_q^{(t)}, \mathbf{t}_r^{(v)} \rangle, \tag{1}$$

where Θ denotes the model parameters, which consist of all latent factors, for instance, $\mathbf{u}_p^{(v)}, \mathbf{u}_p^{(t)} \in \mathbb{R}^K$ denote the latent factors of the user p interacting with the item q like $\mathbf{v}_q^{(u)}$ and the tag r like $\mathbf{t}_r^{(u)}$, respectively, and K is the dimension of latent factors.

The PITF model is optimized with Bayesian Personalized Ranking (BPR) criterion from implicit feedback [15]. Following [14, 16], we call a combination (p, q) a *post*. Given a post (p, q), the BPR assumes that a tag r is preferred over another tag r' if and only if (p, q, r) has been observed and (p, q, r') has not. The core idea is to optimize rankings by considering $(p, q, r, r') \in \mathcal{D}_{pq}$, where

$$\mathcal{D}_{pq} = \{(p,\, q,\, r,\, r') | (p,\, q,\, r) \in \mathcal{D} \wedge (p,\, q,\, r') \notin \mathcal{D}\}.$$

The BPR tries to minimize the following objective function:

$$\mathcal{L}_{\text{BPR}}(\Theta) = \sum_{(p,\, q,\, r,\, r') \in \mathcal{D}_{pq}} - \ln \sigma(\hat{\mathbf{A}}_{pqrr'}(\Theta)) + \lambda \|\Theta\|_F^2, \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function, $\| \cdot \|_F$ is the Frobenius norm, λ is the regularization parameter, and $\hat{\mathbf{A}}_{pqrr'}(\Theta) = \hat{\mathcal{X}}_{pqr}(\Theta) - \hat{\mathcal{X}}_{pqr'}(\Theta)$ for short. The PITF has both linear time complexity and strong ability to capture pairwise interactions among users, tags and items. However, it is vulnerable to random/adversarial perturbations due to its multi-linear assumption (e.g., inner product).

3.3 The ATF Model

Here we propose ATF, a model that combines tensor factorization and adversarial learning to improve the robustness of a tensor model. This builds upon recent work on adversarial machine learning techniques [6, 7, 9, 13], which focus on the potential issue of an unstable learning model. Similar to [9, 13], we inject adversarial perturbations Δ on latent factors to quantify the loss of a tensor model under perturbations on its parameters[1]:

$$\begin{aligned}\hat{\mathcal{X}}_{pqr}(\Theta + \Delta) = \langle \mathbf{u}_p^{(v)} + \Delta\mathbf{u}_p^{(v)}, \mathbf{v}_q^{(u)} + \Delta\mathbf{v}_q^{(u)} \rangle + \langle \mathbf{u}_p^{(t)} + \Delta\mathbf{u}_p^{(t)}, \mathbf{t}_r^{(u)} + \Delta\mathbf{t}_r^{(u)} \rangle \\ + \langle \mathbf{v}_q^{(t)} + \Delta\mathbf{v}_q^{(t)}, \mathbf{t}_r^{(v)} + \Delta\mathbf{t}_r^{(v)} \rangle,\end{aligned} \tag{3}$$

where the perturbation vectors Δ are coupled with their corresponding latent factors, *i.e.*, $\Delta\mathbf{u}_p^{(v)} \in \mathbb{R}^K$ denotes the perturbation vector for latent vector $\mathbf{u}_p^{(v)}$. Moreover, the goal of adversarial perturbations is to cause largest influence on the model, which are also

[1] Note that the term $\langle \mathbf{u}_p^{(v)} + \Delta\mathbf{u}_p^{(v)}, \mathbf{v}_q^{(u)} + \Delta\mathbf{v}_q^{(u)} \rangle$ has not effect in later optimization algorithm because of the BPR loss in Eq. (2), but we still put it here for completion.

known as the worse-case perturbations [7]. Therefore, we find the optimal adversarial perturbations by maximize the BPR loss:

$$\Delta_{adv} = \arg\max_{\Delta} \mathcal{L}_{\text{BPR}}(\hat{\Theta} + \Delta), \qquad s.t. \quad \|\Delta\| \leq \epsilon, \tag{4}$$

where ϵ controls the magnitude of adversarial perturbations, $\|\cdot\|$ denotes the L_2 norm, and $\hat{\Theta}$ is the intermediate model parameters.

To this end, our target is to design a new objective function that is both reasonable for personalized ranking and robust to adversarial perturbations. Formally, we minimize the adversarial BPR loss by combining Eq.(2) and Eq.(4) as follow:

$$\begin{aligned} \mathcal{L}_{\text{ATF}}(\Theta) &= \mathcal{L}_{\text{BPR}}(\Theta) + \alpha \mathcal{L}_{\text{BPR}}(\Theta + \Delta_{adv}), \\ \text{where} \quad \Delta_{adv} &= \arg\max_{\Delta, \|\Delta\| \leq \epsilon} \mathcal{L}_{\text{BPR}}(\hat{\Theta} + \Delta), \end{aligned} \tag{5}$$

where α controls the impact of the adversarial permutations on the model optimization. In the extreme case (e.g., $\alpha = 0$), the proposed ATF becomes the original BPR framework in Eq. (2). Therefore, our ATF can be viewed as a generalization of existing tensor models while considering the robustness of models.

3.4 Optimization

As the intermediate variable Δ maximizes the objective function that is minimized by Θ, the optimization in Eq. (5) can be formulated as a minimax objective function:

$$\Theta^*, \Delta^* = \arg\min_{\Theta} \max_{\Delta, \|\Delta\| \leq \epsilon} \mathcal{L}_{\text{BPR}}(\Theta) + \alpha \mathcal{L}_{\text{BPR}}(\Theta + \Delta), \tag{6}$$

where the optimization of model parameters Θ is the minimizing player and adversarial perturbations Δ is the maximizing player. The two players alternately play the minimax game until convergence. We next provide details to solve the minimax optimization.

Updating Δ: Given a training instance (p, q, r, r'), the adversarial perturbations Δ can be updated by maximizing:

$$\max_{\Delta, \|\Delta\| \leq \epsilon} l_{adv}(\Delta) = -\alpha \ln \sigma(\hat{\mathbf{A}}_{pqrr'}(\hat{\Theta} + \Delta)), \tag{7}$$

where $\hat{\mathbf{A}}_{pqrr'}(\hat{\Theta}+\Delta) = \hat{\mathcal{X}}_{pqr}(\hat{\Theta}+\Delta) - \hat{\mathcal{X}}_{pqr'}(\hat{\Theta}+\Delta)$ and $\hat{\Theta}$ is a constant representing current model parameters. Due to the non-linearity of l_{adv} and the ϵ-constraint in the optimization, we adopt the fast gradient sign method to approximate the objective function around Δ as a linear function [7]. The optimal solution Δ can be achieved by moving the variable towards the direction of its gradient, which can be derived as:

$$\frac{\partial l_{adv}(\Delta)}{\partial \Delta} = -\alpha \cdot \sigma(-\hat{\mathbf{A}}_{pqrr'}(\hat{\Theta} + \Delta)) \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\hat{\Theta} + \Delta)}{\partial \Delta}, \tag{8}$$

With the constraint $\|\Delta\| \leq \epsilon$, we have the optimal solution for Δ_{adv} as:

$$\Delta_{adv} = \epsilon \frac{\Gamma}{\|\Gamma\|} \qquad \text{where} \qquad \Gamma = \frac{\partial l_{adv}(\Delta)}{\partial \Delta}. \tag{9}$$

According to Eq. (9), we can then materialize the gradients of perturbation vectors in Eq. (3) as follows:

$$\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\hat{\Theta} + \Delta)}{\partial \Delta} = \begin{cases} \mathbf{t}_r^{(u)} + \Delta\mathbf{t}_r^{(u)} - \mathbf{t}_{r'}^{(u)} - \Delta\mathbf{t}_{r'}^{(u)}, & \text{if } \Delta = \Delta\mathbf{u}_p^{(t)} \\ \mathbf{t}_r^{(v)} + \Delta\mathbf{t}_r^{(v)} - \mathbf{t}_{r'}^{(v)} - \Delta\mathbf{t}_{r'}^{(v)}, & \text{if } \Delta = \Delta\mathbf{v}_q^{(t)} \\ \mathbf{u}_p^{(t)} + \Delta\mathbf{u}_p^{(t)}, & \text{if } \Delta = \Delta\mathbf{t}_r^{(u)} \\ -\mathbf{u}_p^{(t)} - \Delta\mathbf{u}_p^{(t)}, & \text{if } \Delta = \Delta\mathbf{t}_{r'}^{(u)} \\ \mathbf{v}_q^{(t)} + \Delta\mathbf{v}_q^{(t)}, & \text{if } \Delta = \Delta\mathbf{t}_r^{(v)} \\ -\mathbf{v}_q^{(t)} - \Delta\mathbf{v}_q^{(t)}, & \text{if } \Delta = \Delta\mathbf{t}_{r'}^{(v)} \end{cases} \tag{10}$$

Note that the gradients with respect to $\Delta\mathbf{u}_p^{(v)}$ and $\Delta\mathbf{v}_q^{(u)}$ will be vanished because of the BPR optimization.

Updating Θ: The model parameters Θ can be obtained by minimizing:

$$\min_{\Theta} l_{ATF}(\Theta) = -\ln \sigma(\hat{\mathbf{A}}_{pqrr'}(\Theta)) - \alpha \ln \sigma(\hat{\mathbf{A}}_{pqrr'}(\Theta + \Delta_{adv})) + \lambda \|\Theta\|_F^2, \tag{11}$$

where Δ_{adv} is a constant computed from Eq. (9). The derivative with respect to Θ is:

$$\begin{aligned} \frac{\partial l_{ATF}(\Theta)}{\partial \Theta} = &- \sigma(-\hat{\mathbf{A}}_{pqrr'}(\Theta)) \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \Theta} + 2\lambda\Theta \\ &- \alpha \cdot \sigma(-\hat{\mathbf{A}}_{pqrr'}(\Theta + \Delta_{adv})) \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta + \Delta_{adv})}{\partial \Theta}, \end{aligned} \tag{12}$$

then we can use stochastic gradient descent update rule:

$$\Theta \leftarrow \Theta - \eta \frac{\partial l_{ATF}(\Theta)}{\partial \Theta} \tag{13}$$

where η is the learning rate. We then compute the derivatives of latent factors in Eq. (3) by materializing $\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \Theta}$ in Eq. (12) as:

$$\begin{aligned} &\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \mathbf{u}_p^{(t)}} = \mathbf{t}_r^{(u)} - \mathbf{t}_{r'}^{(u)}, \ \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \mathbf{t}_r^{(u)}} = \mathbf{u}_p^{(t)}, \ \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \mathbf{t}_r^{(v)}} = \mathbf{v}_q^{(t)}, \\ &\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \mathbf{v}_q^{(t)}} = \mathbf{t}_r^{(v)} - \mathbf{t}_{r'}^{(v)}, \ \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \mathbf{t}_{r'}^{(u)}} = -\mathbf{u}_p^{(t)}, \ \frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta)}{\partial \Delta\mathbf{t}_{r'}^{(v)}} = -\mathbf{v}_q^{(t)}, \end{aligned} \tag{14}$$

and $\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta+\Delta_{adv})}{\partial \Theta}$ as:

$$\frac{\partial \hat{\mathbf{A}}_{pqrr'}(\Theta + \Delta_{adv})}{\partial \Theta} = \begin{cases} \mathbf{t}_r^{(u)} + \Delta_{adv}\mathbf{t}_r^{(u)} - \mathbf{t}_{r'}^{(u)} - \Delta_{adv}\mathbf{t}_{r'}^{(u)}, & \text{if } \Theta = \mathbf{u}_p^{(t)} \\ \mathbf{t}_r^{(v)} + \Delta_{adv}\mathbf{t}_r^{(v)} - \mathbf{t}_{r'}^{(v)} - \Delta_{adv}\mathbf{t}_{r'}^{(v)}, & \text{if } \Theta = \mathbf{v}_q^{(t)} \\ \mathbf{u}_p^{(t)} + \Delta_{adv}\mathbf{u}_p^{(t)}, & \text{if } \Theta = \mathbf{t}_r^{(u)} \\ -\mathbf{u}_p^{(t)} - \Delta_{adv}\mathbf{u}_p^{(t)}, & \text{if } \Theta = \mathbf{t}_{r'}^{(u)} \\ \mathbf{v}_q^{(t)} + \Delta_{adv}\mathbf{v}_q^{(t)}, & \text{if } \Theta = \mathbf{t}_r^{(v)} \\ -\mathbf{v}_q^{(t)} - \Delta_{adv}\mathbf{v}_q^{(t)}, & \text{if } \Theta = \mathbf{t}_{r'}^{(v)} \end{cases} \tag{15}$$

Similarly, the gradients with respect to $\mathbf{u}_p^{(v)}$ and $\mathbf{v}_q^{(u)}$ will be vanished. According to above analysis, we summarize the algorithm for solving the minimax optimization Eq. (6) in Algorithm 1.

Algorithm 1: Learning algorithm for ATF

```
Input: Training dataset $\mathcal{D}$, dimension of latent factor $K$, noise
       level $\epsilon$, regularizer $\lambda$ and $\alpha$, and learning rate $\eta$;
Output: Model parameters $\Theta$;
Initialize $\Theta$ from original BPR by solving Eq. (2);
Initialize $\Delta$ randomly, such that $\|\Delta\| \leq \epsilon$;
repeat
    Draw $(p, q, r, r')$ uniformly from $\mathcal{D}_{pq}$;
    Update $\Delta_{adv}$ by Eq. (9);
    Update $\Theta$ by Eq. (13);
until Convergence
return $\Theta$
```

4 EXPERIMENTS

4.1 Dataset and Baselines

We consider two public datasets from HetRec 2011[2]:

[2] https://grouplens.org/datasets/hetrec-2011/

Table 1: Dataset statistics (after preprocessing)

Dataset	User $\lvert\mathcal{U}\rvert$	Item $\lvert\mathcal{I}\rvert$	Tag $\lvert\mathcal{T}\rvert$	Triple $\lvert\mathcal{D}\rvert$
MovieLens	693	2,634	1,582	30,830
Last.fm	2,917	1,853	2,045	219,702

- **MovieLens** is published by GroupLeans research group to analyze the relationships among users, tags and movies, which includes 2, 113 users, 13, 222 tags and 10, 197 movies (regarded as "items" in this study).
- **Last.fm** is an online music system that contains each user's musical taste by recording the tracks the user listens to. The original dataset consists of 1, 892 users, 11, 946 tags and 17, 632 artists.

Follow data preprocessing in [14, 16, 18], we use 5-core for the MovieLens dataset and 10-core for the Last.fm dataset. Table 1 lists statistics of our dense datasets. For each post (p, q) in the set $\mathcal{D}$, there exist at least k triples (p, q, r) in the observed data since we adopt k-core processing. We randomly withhold one triple (p, q, r_v) for validation, and another triple (p, q, r_t) for testing. All remaining triples are used for training.

Baselines: We compare our method with standard tensor models in tag recommendations:

- **CANDECOMP/PARAFAC (CP)** [11]: a multi-linear model that expresses a tensor as outer product of latent factors.
- **HOSVD** [18]: a tensor based method based on the higher order singular value decomposition method.
- **PITF** [16]: a pairwise interaction tensor factorization, which is the basic model our ATF is built on.

For a fair comparison, the size of latent factors, *i.e.,* K in Eq. (1), are all manually set with $K = 60$. For both PITF and ATF, we vary the value of BPR regularizer λ within $\{0.001, 0.01, 0.1, 1\}$, the value of learning rate η within $\{0.005, 0.01, 0.05\}$. For ATF, we tune error level ϵ in $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$ and the adversarial regularizer α in $\{0.001, 0.01, 0.1, 1, 10, 100\}$. Since the size of data instances in $\mathcal{D}$ is huge, we randomly sample an unobserved tag r' to pair with an observed triple (p, q, r) to construct $\mathcal{D}_{pq}$. We tune the hyper-parameters via the validation set and report the performance on the test set. We use the common evaluation scheme of F1-measure for top-N recommendations, which is the overall evaluation on the average recall and precision. The definition of F1-measure can be found in [14, 16] for details. The experiments are repeated five times and the average results are reported in this study.

4.2 Recommendation Performance

Figure 1 shows the top-N (e.g., $N = 2, 4, 6, 8, 10$) ranking performance of all methods on the two datasets in terms of F1-scores. From the results, we can see that ATF outperforms existing baselines on both datasets. Furthermore, we observe that ATF and PITF achieve better prediction accuracy than HOSVD and CP since they explicitly consider any pairwise interactions among users, tags and items. Compared to PITF, ATF has an average gain of 6.98% for the MovieLens dataset and an average gain of 6.15% for the Last.FM dataset. This could be ascribed to the ability of the proposed ATF in handling adversarial perturbations during the training step, which improves its eventual performance in context-aware recommendations

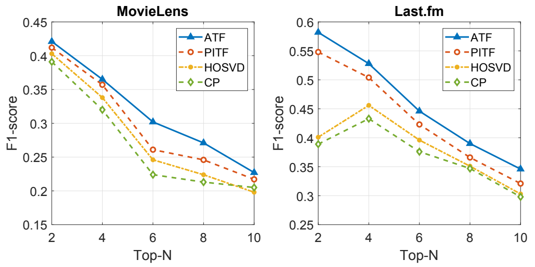

Figure 1: Performance comparison of different tensor models in top-N recommendations

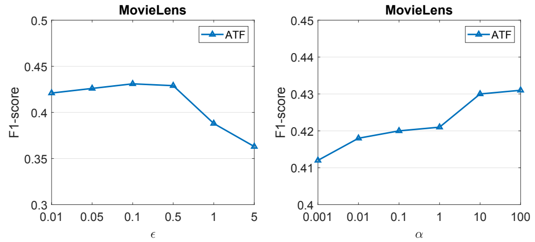

Figure 2: The impacts of the ϵ and α for top-2 recommendations for the MovieLens dataset.

The impacts of ϵ and α: Compared to PITF, our proposed APR model introduces two additional hyper-parameters ϵ and α to control the perturbation level and the strength of adversarial regularizer, respectively. Here we only study their impacts on the MovieLens dataset due to space limitation. In the experiments, we fix one parameter, and study the impacts of the other. To study ϵ,we fix $\alpha = 10$. Similarly, we fix $\epsilon = 0.5$ when studying α. Figure 2 shows the top-2 ranking performance. We can see that ATF is mostly stable when $\epsilon \leq 0.5$. When ϵ is too large, the performance drops dramatically, which means that certain large magnitude of adversarial perturbations are injected and maliciously attack the model parameters during training step. For adversarial regularizer α, a relatively high accuracy can achieved with a large value of α, which shows the importance of adversarial learning mechanism.

5 CONCLUSIONS AND FUTURE WORK

In this work, we consider the robustness of tensor-based models for context-aware recommendations, and design a new method ATF, which combines tensor factorization and adversarial learning. We also develop a learning algorithm to solve our minimax optimization. Empirically, extensive results on two real-word datasets demonstrate the effectiveness of our method. In the future, we plan to further investigate the problem of incorporating more contextual factors to better understand the users' behaviors and interests, such as time, location, and users' social networks.

Acknowledgements: This work has been supported in part by NIH grants R03HG008632, NSF CCF1815139, and a Faculty Investment Fund from CWRU.

REFERENCES

[1] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *RecSys*.

[2] Huiyuan Chen and Jing Li. 2017. Learning multiple similarities of users and items in recommender systems. In *ICDM*.

[3] Huiyuan Chen and Jing Li. 2018. DrugCom: Synergistic Discovery of Drug Combinations Using Tensor Decomposition. In *ICDM*.

[4] Huiyuan Chen and Jing Li. 2019. Modeling Relational Drug-Target-Disease Interactions via Tensor Factorization with Multiple Web Sources. In *WWW*.

[5] Gene H Golub and Charles F Van Loan. 2012. *Matrix computations*. Vol. 3. JHU press.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

[8] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *RecSys*.

[9] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *SIGIR*.

[10] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*.

[11] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* (2009).

[12] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. 2016. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *CVPR*.

[13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *CVPR*.

[14] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD*.

[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.

[16] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*.

[17] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. 2012. TFMAP: optimizing MAP for top-n context-aware recommendation. In *SIGIR*.

[18] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2008. Tag recommendations based on tensor dimensionality reduction. In *RecSys*.

[19] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*.

[20] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*.

[21] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *WWW*.