Fairness-Aware Tensor-Based Recommendation

Ziwei Zhu, Xia Hu, and James Caverlee
Department of Computer Science and Engineering, Texas A&M University
{zhuziwei,hu,caverlee}@tamu.edu

ABSTRACT

Tensor-based methods have shown promise in improving upon traditional matrix factorization methods for recommender systems. But tensors may achieve improved recommendation quality while worsening the fairness of the recommendations. Hence, we propose a novel fairness-aware tensor recommendation framework that is designed to maintain quality while dramatically improving fairness. Four key aspects of the proposed framework are: (i) a new sensitive latent factor matrix for isolating sensitive features; (ii) a sensitive information regularizer that extracts sensitive information which can taint other latent factors; (iii) an effective algorithm to solve the proposed optimization model; and (iv) extension to multi-feature and multi-category cases which previous efforts have not addressed. Extensive experiments on real-world and synthetic datasets show that the framework enhances recommendation fairness while preserving recommendation quality in comparison with state-of-the-art alternatives.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

recommender systems; fairness-aware; tensor completion

ACM Reference Format:
Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271795

1 INTRODUCTION

Recommenders are essential conduits: they shape the media we consume, the jobs we seek, and the friendships and professional contacts that form our social circles. And yet, recommenders may be subject to algorithmic bias that can lead to negative consequences in the kinds of recommendations that are made. For example, job recommenders can target women with lower-paying jobs than equally-qualified men [6]. News recommenders can favor particular political ideologies over others [2]. And even ad recommenders can exhibit racial discrimination [26].

Overcoming such algorithmic bias has been of keen interest in classification tasks (e.g., recidivism prediction, loan approval) [5, 22, 25, 30, 31], but has only recently been on the rise for recommender systems [12, 13, 28, 29]. Recommender-based approaches have typically focused on *fairness*, where the goal is to maintain some level of neutrality in recommendation, e.g., balancing male vs. female or old vs. young. While encouraging, most existing approaches make a number of limiting assumptions: (i) focusing on two-dimensional matrix factorization that has been the cornerstone of recommender research in the past ten years [11, 19, 21]; (ii) assuming there is only a single binary case (e.g., male vs. female); and (iii) trading-off considerable recommendation quality for improving the fairness characteristics of the recommender.

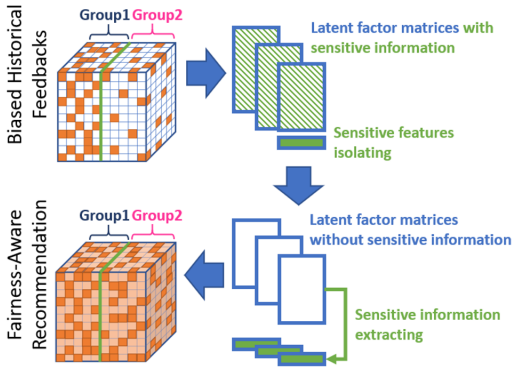


Figure 1: Overview of FATR: sensitive features are isolated (top right), then sensitive information is extracted (bottom right), resulting in fairness-aware recommendation.

In contrast, we aim in this paper to create a new *tensor-based* framework that can overcome these limitations for implicit recommendation (i.e. where implicit feedback is available, but no explicit ratings). Tensors, as n-dimensional generalizations of matrices, have shown great promise across a variety of data mining and analytics tasks – e.g., [8, 17, 23, 24] – where their multi-aspect models naturally fit domains that go beyond two dimensions. Recommenders, in particular, are well-suited for tensors that can capture multi-way interactions among users, items, and contexts (e.g., time, location). But there are key challenges: How can we model sensitive attributes (e.g., age, gender) in a tensor-based recommender? How can we minimize the impact of these sensitive attributes on recommendations, which can be correlated with non-sensitive attributes [16, 31])? How can we build an optimization model for this problem and efficiently solve it? And can such efforts maintain recommendation quality while improving fairness?

Toward answering these challenges, this paper proposes a novel **F**airness-**A**ware **T**ensor-based **R**ecommendation framework called **FATR**. The overview is illustrated in Figure 1. The intuition of the proposed framework is that the latent factor matrices of the tensor completion model contain latent information related to the sensitive attributes, which introduces the unfairness. Therefore, by *isolating* and then *extracting* the sensitive information from the

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271795

latent factor matrices, we may be able to improve the fairness of the recommender itself. Concretely, we propose (i) a new *sensitive latent factor matrix* for isolating sensitive features; (ii) a sensitive information regularizer that extracts sensitive information which can taint other latent factors; and (iii) an effective algorithm to solve the proposed optimization model.

In sum, the contributions of this paper are as follows.

- First, FATR is built on a tensor foundation that can analyze multiple aspects simultaneously, promising potentially better recommendation quality than matrix-based approaches, while also supporting traditional two-dimensional data (since tensors are generalizations of matrices).
- Second, moving beyond binary sensitive features, FATR supports multi-feature cases with multisided features (e.g., recommendation where both age of items and gender of users are considered sensitive) and multi-category cases (e.g., where the sensitive attribute can take on multiple values like Low, Medium, and High) which are challenging for traditional regularization-based approaches [14, 29].
- Finally, we empirically show that FATR can provide recommendation quality on par with traditional (unfair) recommenders while significantly improving the fairness of recommendations, and does so better than state-of-the-art alternatives.

2 PRELIMINARIES

In this section, we first introduce the notations used in this paper and the basics of tensor-based recommendation, then we discuss fairness in recommendation.

2.1 Notations

Notations and definitions in this paper are presented as follows. Tensors are denoted by Euler script letters like $\mathcal{X}$, matrices are denoted by boldface uppercase letters like $\mathbf{A}$, and vectors are denoted by boldface lowercase letters like $\mathbf{a}$. The $[i_1, \ldots, i_N]$ entry of the tensor $\mathcal{X}$ is denoted as $\mathcal{X}[i_1, \ldots, i_N]$. We denote the pseudo inverse, transpose, and Frobenius norm of a matrix $\mathbf{A}$ respectively by $\mathbf{A}^\dagger$, $\mathbf{A}^\top$, and $\|\mathbf{A}\|_\mathrm{F}$. Notation $[\![\cdot]\!]$ represents the Kruskal operator. Notations $\odot$, $\circledast$, and $\circ$ denote the Khatri-Rao product, Hadamard product, and vector outer product, respectively. Besides, we use the syntax similar to Python to denote the matrix slicing operation (the index starts from 1), for example $\mathbf{A}[:, 2:]$ denotes the matrix $\mathbf{A}$ without the first column. And we use $[\mathbf{A}\ \ \mathbf{B}]$ to present the horizontal matrices concatenating operation. The main symbols and operations are listed in Table 1. More details about tensor calculations can be found in [18].

2.2 Tensor-Based Recommendation

Matrix factorization is the foundation of many modern recommenders [20]. These matrix factorization methods estimate missing ratings by uncovering latent features of users and items. Building on these user-item interactions, *tensor-based* methods have been growing in appeal recently since they can naturally model multi-way (or multi-aspect) interactions [8, 17, 23, 24]. For example, a 3-order tensor could represent users, items, and time of day. Additional contexts can lead to an N-way tensor. And, of course, the classic user-item problem can be viewed as a 2-way tensor.

Notations	Definitions
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$	N^{th}-order tensor
$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (\prod_{i \neq n}^{N} I_i)}$	Mode-n unfolding matrix of tensor $\mathcal{X}$
$[\![\cdot]\!]$	Kruskal operator, e.g., $\mathcal{X} \approx [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]$
$\odot$	Khatri-Rao product
$\circledast$	Hadamard product
$\circ$	Vector outer product
$(\mathbf{A}_k)^{\odot k \neq n}$	$\mathbf{A}_N \odot \ldots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \ldots \odot \mathbf{A}_1$
$\mathbf{A}[:, i:j]$	Matrix slicing operation (index starts from 1)
$[\mathbf{A}\ \ \mathbf{B}]$	Matrices concatenating operation (horizontal)

Table 1: Main symbols and operations.

Formally, given an N-order tensor $\mathcal{T}$ representing the users, items, and multiple aspects related to the items, the basic tensor-based recommendation model can be defined as:

$$\begin{aligned} \underset{\mathcal{X}, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathcal{X} - [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N]\!]\|_\mathrm{F}^2 \\ \text{subject to} \quad & \mathbf{\Omega} \circledast \mathcal{X} = \mathcal{T}, \end{aligned}$$

where $\mathcal{X}$ denotes the complete preferences of users, $\mathcal{T}$ denotes the observations, $\mathbf{\Omega}$ is a non-negative indicator tensor with the same size as $\mathcal{X}$ with $\mathbf{\Omega}[i_1, \ldots, i_N] = 1$ indicating that we observe the preference, otherwise $\mathbf{\Omega}[i_1, \ldots, i_N] = 0$, $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N$ are the latent factor matrices of all the modes of the tensor.

The objective function can be written in the unfolding form so that it can be solved by optimization algorithms, as follows:

$$\begin{aligned} \underset{\mathcal{X}, \mathbf{A}_1, \ldots, \mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathbf{X}_{(n)} - \mathbf{A}_n[(\mathbf{A}_k)^{\odot k \neq n}]^\top\|_\mathrm{F}^2 \\ \text{subject to} \quad & \Omega_{(n)} \circledast \mathbf{X}_{(n)} = \mathbf{T}_{(n)}, \end{aligned} \tag{1}$$

where $\Omega_{(n)}$ is the mode-n unfolding of the indicator tensor $\mathbf{\Omega}$, $\mathbf{T}_{(n)}$ is the mode-n unfolding of the tensor $\mathcal{T}$, and $\mathbf{X}_{(n)}$ is the mode-n unfolding of the tensor $\mathcal{X}$. To solve this basic recommendation by tensor completion, we can use Alternating Least Squares (ALS), which optimizes every latent factor matrix by linear least squares in each iteration. The update rule is:

$$\widehat{\mathbf{A}_n} \leftarrow \mathbf{X}_{(n)}[[(\mathbf{A}_k)^{\odot k \neq n}]^\top]^\dagger,$$

where $\widehat{\mathbf{A}_n}$ is the updated latent factor matrix of $\mathbf{A}_n$.

2.3 Fairness in Recommendation

Such a tensor-based approach has no notion of fairness. Here, we assume that there exists a *sensitive attribute* for one mode of the tensor, and this mode is a *sensitive mode*. For example, the sensitive attribute could correspond to gender, age, ethnicity, location, or other domain-specific attributes of users or items in the recommenders. The feature vectors of the sensitive attributes are called the *sensitive features*. Further, we call all the information related to the sensitive attributes as *sensitive information*, and note that attributes other than the sensitive attributes can also contain sensitive information [16, 31]. While there are emerging debates about what constitutes algorithmic fairness [5], we adopt the commonly used notion of *statistical parity*. Statistical parity encourages a recommender to ensure similar probability distributions for both the dominant group and the protected group as defined by the sensitive attributes. Formally, we denote the sensitive attribute as a random variable S, and the preference rating in the recommender system as a random variable R. Then we can formulate fairness

as $\mathbf{P}[R] = \mathbf{P}[R|S]$, i.e. the preference rating is independent of the sensitive attribute. This statistical parity means that the recommendation result should be unrelated to the sensitive attributes. For example, a job recommender should recommend similar jobs to men and women with similar profiles. Note that some recent works [9, 28, 29] have argued that statistical parity may be overly strict, resulting in poor utility to end users. Our work here aims to achieve comparable utility to non-fair approaches, while providing stronger fairness.

3 FAIRNESS-AWARE TENSOR-BASED RECOMMENDATION

Given this notion of fairness, we turn in this section to the design of a novel **F**airness-**A**ware **T**ensor-based **R**ecommendation framework (**FATR**) – as illustrated in Figure 2. The intuition of the proposed framework is that the latent factor matrices of the tensor completion model contain latent information related to the sensitive attributes, which introduces the unfairness. Therefore, by *isolating* and then *extracting* the sensitive information from the latent factor matrices, we may be able to improve the fairness of the recommender itself.

In the rest of this section, we aim to address four key questions: (i) How can we represent (and ultimately isolate) the sensitive attributes in the tensor completion model? (ii) How do we extract all the sensitive information into the isolated explicit representation? (iii) How can we eliminate the extracted sensitive information from the tensor completion model? and (iv) How do we solve the new fairness-aware recommendation model? In the following, we address these questions in turn. We focus in this section on a single binary sensitive attribute for mode-n (e.g., gender). In Section 4, we will generalize to consider multi-feature and multi-category cases.

3.1 Isolating Sensitive Features

In conventional tensor completion, the sensitive features will mingle with other features and distribute over different dimensions in the latent factor matrices, which makes it difficult to extract them. For example, a 3-way tensor of user-expert-topic can be factorized into three latent factor matrices [8], where the feature vector of a sensitive attribute for the experts like gender is mixed with other features and represented by the latent factors, which means that the sensitive information hides in the expert latent factor matrix.

We propose to first isolate the impact of the sensitive attribute by plugging the sensitive features into the latent factor matrix. For instance, in our user-expert-topic example, we can create one vector $\mathbf{s}_0$ with 1 representing male and 0 representing female, and another vector $\mathbf{s}_1$ with 1 indicating female and 0 indicating male. $\mathbf{s}_0$ and $\mathbf{s}_1$ together form a matrix, denoted as *Sensitive Features* $\mathbf{S}$. We put $\mathbf{S}$ to the last two columns of the latent factor matrix of sensitive mode mode-n. Then we construct a new *sensitive latent factor matrix*:

Definition *(Sensitive Latent Factor Matrix).* Given the latent factor matrix $\mathbf{A}_n \in \mathbb{R}^{d_n \times r}$ of the sensitive mode mode-n, where r is the dimension of the latent factors and d_n is the number of entities of the mode-n. We split $\mathbf{A}_n$ horizontally into two parts: matrix $\mathbf{A}'_n \in \mathbb{R}^{d_n \times (r-2)}$ and $\mathbf{A}''_n \in \mathbb{R}^{d_n \times 2}$. If $\mathbf{A}''_n$ is forced to take the same values as the sensitive features $\mathbf{S} \in \mathbb{R}^{d_n \times 2}$, then the new matrix $\widetilde{\mathbf{A}_n} = [\mathbf{A}'_n \ \ \mathbf{S}]$ is called sensitive latent factor matrix.

The matrix $\mathbf{A}'_n$ represents the *non-sensitive dimensions*, while $\mathbf{A}''_n$ represents the *sensitive dimensions* (where the corresponding dimensions in other non-sensitive factor latent matrices are also called sensitive dimensions). Thus, sensitive dimensions of the sensitive latent factor matrix will take the same values of the sensitive features. In this way, we can explicitly represent the sensitive attributes and isolate them from non-sensitive attributes in the latent factor matrix. Hence, we can update the tensor-based recommender in Section 2.2 with the following objective function:

$$\begin{aligned} \underset{\mathcal{X}, \mathbf{A}_1, \ldots, \widetilde{\mathbf{A}_n}, \ldots, \mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathcal{X} - [\![\mathbf{A}_1, \ldots, \widetilde{\mathbf{A}_n}, \ldots, \mathbf{A}_N]\!]\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \boldsymbol{\Omega} \circledast \mathcal{X} = \mathcal{T}, \\ & \widetilde{\mathbf{A}_n} = [\mathbf{A}'_n \ \ \mathbf{A}''_n], \\ & \mathbf{A}''_n = \mathbf{S}. \end{aligned}$$

3.2 Extracting Sensitive Information

By isolating the sensitive features, we provide a first step toward improving the fairness of the recommender. But there may still be sensitive information that resides in non-sensitive dimensions. To extract this remaining sensitive information, we propose an additional constraint that the non-sensitive dimensions should be orthogonal to the sensitive dimensions in the sensitive latent factor matrix based on the following theorem.

Theorem. If one non-sensitive dimension is not perpendicular to all the sensitive dimensions, then this dimension is related to the sensitive attribute.

Proof. Regarding all dimensions in the sensitive latent factor matrix as vectors in a high dimensional space. If the angle between one non-sensitive dimension vector $\mathbf{v}$ and the plane p_{s_1,s_2} decided by sensitive features $\mathbf{s}_1$ and $\mathbf{s}_2$ is not 90°, then $\mathbf{v}$ can be resolved into two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ on the same directions as $\mathbf{s}_1$ and $\mathbf{s}_2$, and another vector $\mathbf{v}_3$ perpendicular to p_{s_1,s_2}. Therefore, $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3$, and $\mathbf{v}_1$ and $\mathbf{v}_2$ can be merged into $\mathbf{s}_1$ and $\mathbf{s}_2$ when reconstructing the tensor as shown in Equation (2), which changes the values of sensitive dimensions, i.e., this latent factor represented by dimension $\mathbf{v}$ is related to the sensitive attribute.

$$\begin{aligned} \mathcal{X} \approx \ & \mathbf{a}_1^{(1)} \circ \mathbf{a}_2^{(1)} \circ \mathbf{a}_3^{(1)} + \ldots + \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)} \circ \mathbf{v} \\ & + \mathbf{a}_1^{(r-1)} \circ \mathbf{a}_3^{(r-1)} \circ \mathbf{s}_1 + \mathbf{a}_1^{(r)} \circ \mathbf{a}_2^{(r)} \circ \mathbf{s}_2 \\ = \ & \mathbf{a}_1^{(1)} \circ \mathbf{a}_2^{(1)} \circ \mathbf{a}_3^{(1)} + \ldots + \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)} \circ \mathbf{v}_3 \\ & + l_1 \cdot \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)} \circ \mathbf{s}_1 + l_1 \cdot \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)} \circ \mathbf{s}_2 \\ & + \mathbf{a}_1^{(r-1)} \circ \mathbf{a}_3^{(r-1)} \circ \mathbf{s}_1 + \mathbf{a}_1^{(r)} \circ \mathbf{a}_2^{(r)} \circ \mathbf{s}_2 \\ = \ & \mathbf{a}_1^{(1)} \circ \mathbf{a}_2^{(1)} \circ \mathbf{a}_3^{(1)} + \ldots + \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)} \circ \mathbf{v}_3 \\ & + (\mathbf{a}_1^{(r-1)} \circ \mathbf{a}_2^{(r-1)} + l_1 \cdot \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)}) \circ \mathbf{s}_1 \\ & + (\mathbf{a}_1^{(r)} \circ \mathbf{a}_2^{(r)} + l_2 \cdot \mathbf{a}_1^{(r-2)} \circ \mathbf{a}_2^{(r-2)}) \circ \mathbf{s}_2, \end{aligned} \tag{2}$$

where $\mathbf{a}_1^{(1\ldots r)}$, $\mathbf{a}_2^{(1\ldots r)}$, and $\mathbf{a}_3^{(1\ldots r)}$ are the columns in the three latent factor matrices, $l_1$is the scale coefficient between $\mathbf{s}_1$ and $\mathbf{v}_1$ so that $l_1 \cdot \mathbf{s}_1 = \mathbf{v}_1$, and l_2 is from $l_2 \cdot \mathbf{s}_2 = \mathbf{v}_2$.

After extracting the sensitive information, all the sensitive information is gathered in the isolated sensitive dimensions. Then we can have a new objective function for the tensor completion as

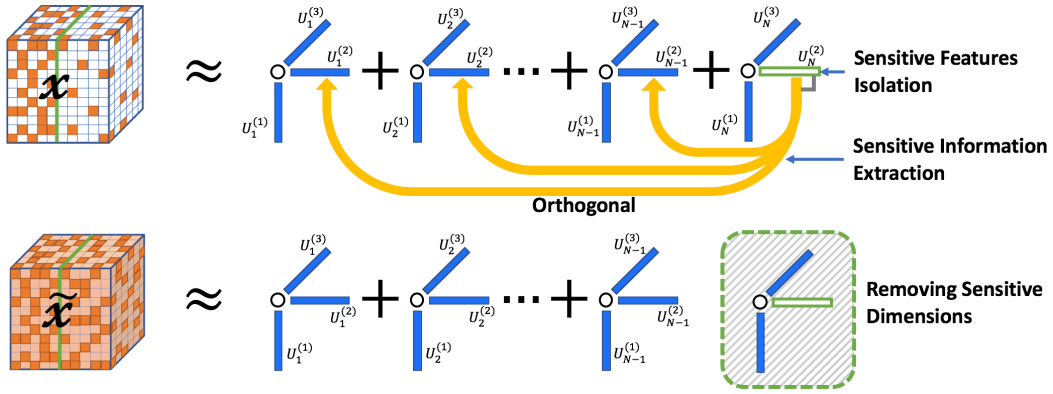


Figure 2: FATR isolates sensitive features in the latent matrix with non-sensitive dimensions orthogonal to them and eliminates the sensitive information by removing the sensitive dimensions. $\mathcal{X}$ is the tensor with bias, and $\widetilde{\mathcal{X}}$ is the fairness-enhanced recommendation tensor.

shown in Equation (3).

$$\begin{aligned} \underset{\mathcal{X},\mathbf{A}_1,\ldots,\mathbf{A}'_n,\ldots,\mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathcal{X} - [\![\mathbf{A}_1,\ldots,\widetilde{\mathbf{A}_n},\ldots,\mathbf{A}_N]\!]\|_{\mathrm{F}}^2 \\ & \quad + \frac{\lambda}{2}\|\mathbf{A}''^{\top}_n \mathbf{A}'_n\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \boldsymbol{\Omega} \circledast \mathcal{X} = \mathcal{T}, \\ & \widetilde{\mathbf{A}_n} = [\mathbf{A}'_n \;\; \mathbf{A}''_n], \\ & \mathbf{A}''_n = \mathbf{S}, \end{aligned} \tag{3}$$

where $\frac{\lambda}{2}\|\mathbf{A}''^{\top}_n \mathbf{A}'_n\|_{\mathrm{F}}^2$ is the orthogonal constraint term, λ is the trade-off parameter, $\frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2$ is the L2-norm constraint to the norms of the latent factor matrices so that the minimizing of $\frac{\lambda}{2}\|\mathbf{A}''^{\top}_n \mathbf{A}'_n\|_{\mathrm{F}}^2$ is because the cosine angles are close to zero rather than because the norms of columns in $\mathbf{A}''_n$ or $\mathbf{A}'_n$ are small (if it is this case, norms of other latent factor matrices will get larger, which will increase the value of the term $\frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2$), γ is the trade-off parameter of this L2-norm term.

3.3 Fairness-Aware Recommendation

After the above two steps, we can get the new latent factor matrices $\mathbf{A}_1,\ldots,\widetilde{\mathbf{A}_n},\ldots,\mathbf{A}_N$, whose sensitive dimensions hold features exclusively related to the sensitive attributes. And their non-sensitive dimensions are decoupled from the sensitive attributes. Thus, we can derive the fairness-enhanced recommendation by combining these matrices after removing their sensitive dimensions as:

$$\widetilde{\mathcal{X}} \leftarrow [\![\mathbf{A}'_1,\ldots,\mathbf{A}'_n,\ldots,\mathbf{A}'_N]\!],$$

where $\widetilde{\mathcal{X}}$ is the fairness-enhanced tensor completion result, and $\mathbf{A}'_1,\ldots,\mathbf{A}'_n,\ldots,\mathbf{A}'_N$ are the non-sensitive dimensions of the latent factor matrices (i.e. the first $r-2$ columns in $\mathbf{A}_1,\ldots,\widetilde{\mathbf{A}_n},\ldots,\mathbf{A}_N$).

3.4 Optimization Algorithms

To solve the optimization problem in Equation (3), we need to first rewrite the objective function to be the unfolding matrix form. For the sensitive mode mode-n, the unfolding form is in Equation (4).

Algorithm 1: FATR Solver

Input: $\mathcal{T}$, $\boldsymbol{\Omega}$, r, $\mathbf{S}$, n, α, λ, γ, tol;
Output: $\widetilde{\mathcal{X}}$, $\{\mathbf{A}_i\}_{i=1}^{N}$

Randomly Initialize $\{\mathbf{A}_i \in \mathbb{R}^{I_i \times r}\}_{i=1}^{N}$;
repeat
  for $i = 1 : N$ **do**
    if $i == n$ **then**
      Update $\mathbf{A}'_n$ using (7);
    else
      Update $\mathbf{A}_i$ using (6);
  Update $\mathbf{A}'_n \leftarrow \mathbf{A}'_n - \alpha\frac{\partial \mathcal{F}}{\partial \mathbf{A}'_n}$;
  Form $\widetilde{\mathbf{A}_n} \leftarrow [\mathbf{A}'_n \;\; \mathbf{S}]$;
  Update $\mathcal{X} \leftarrow \mathcal{T} + \boldsymbol{\Omega} \circledast [\![\mathbf{A}_1,\ldots,\widetilde{\mathbf{A}_n},\ldots,\mathbf{A}_N]\!]$;
until $\|\mathcal{X}_{pre} - \mathcal{X}\|_F / \|\mathcal{X}_{pre}\|_F < tol$;
Update $\widetilde{\mathcal{X}} \leftarrow [\![\mathbf{A}'_1,\ldots,\mathbf{A}'_n,\ldots,\mathbf{A}'_N]\!]$;

$$\begin{aligned} \underset{\mathcal{X},\mathbf{A}_1,\ldots,\mathbf{A}'_n,\ldots,\mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathbf{X}_{(n)} - \mathbf{A}''_n(\mathbf{B}''_n)^{\top} - \mathbf{A}'_n(\mathbf{B}'_n)^{\top}\|_{\mathrm{F}}^2 \\ & \quad + \frac{\lambda}{2}\|\mathbf{A}''^{\top}_n \mathbf{A}'_n\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \boldsymbol{\Omega}_{(n)} \circledast \mathbf{X}_{(n)} = \mathbf{T}_{(n)}, \\ & \mathbf{B}_n = [(\mathbf{A}_k)^{\odot_{k\neq n}}], \\ & \mathbf{B}'_n = \mathbf{B}_n[:, : r-2], \\ & \mathbf{B}''_n = \mathbf{B}_n[:, r-1 :], \\ & \mathbf{A}''_n = \mathbf{S}, \end{aligned} \tag{4}$$

where $\mathbf{B}_n$ is the result of the Khatri-Rao product of all the latent factor matrices without mode-n, $\mathbf{B}'_n$ is the first $r-2$ dimensions of $\mathbf{B}_n$, and $\mathbf{B}''_n$ is the last 2 dimensions of $\mathbf{B}_n$.

For non-sensitive modes (denoted as m), the unfolding objective function is shown in Equation (5).

$$\underset{\mathcal{X},\mathbf{A}_1,\ldots,\mathbf{A}_N}{\text{minimize}} \quad \mathcal{L} = \|\mathbf{X}_{(m)} - \mathbf{A}_m[(\mathbf{A}_k)^{\odot_{k\neq m}}]^\top\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2 \tag{5}$$

$$\text{subject to} \quad \Omega_{(m)} \circledast \mathbf{X}_{(m)} = \mathbf{T}_{(m)}, \ \ \mathbf{A}_n'' = \mathbf{S}.$$

Equation (4) cannot be solved by ALS because of $\frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$, but Equation (5) can be solved by ALS because $\frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$ is a constant term for non-sensitive modes. We can use Gradient Descent to solve them together, but its performance is not as good as ALS for tensor completion task. However, if we can separate $\frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$ from the objective function and optimize it alone, we can efficiently and effectively solve the problem. Thus, we propose a hybrid optimization algorithm which treats the sensitive and non-sensitive modes differently. It follows the ALS rule to update the non-sensitive modes in each iteration. For the sensitive mode mode-n, we first use ALS to update $\mathbf{A}_n'$ with $\frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$ being considered as a constant term, and then use Gradient Descent to update $\mathbf{A}_n'$ again only to minimize $\frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$. The update rule for the non-sensitive modes is defined in rule (6), and the first ALS step for the sensitive mode mode-n uses update rule (7).

$$\widetilde{\mathbf{A}_m} \leftarrow \mathbf{X}_{(m)}(\mathbf{A}_k)^{\odot_{k\neq m}}[\gamma\mathbf{I} + [(\mathbf{A}_k)^{\odot_{k\neq m}}]^\top(\mathbf{A}_k)^{\odot_{k\neq m}}]^\dagger, \tag{6}$$

$$\widetilde{\mathbf{A}_n}' \leftarrow [\mathbf{X}_{(n)} - \mathbf{A}_n''(\mathbf{B}_n'')^\top]\mathbf{B}_n'[\gamma\mathbf{I} + (\mathbf{B}_n')^\top\mathbf{B}_n']^\dagger, \tag{7}$$

where $\widetilde{\mathbf{A}_m}$ is the updated non-sensitive latent factor matrix, $\widetilde{\mathbf{A}_n}'$ is the updated non-sensitive dimensions of the sensitive latent factor matrix, $\mathbf{I}$ is an identity matrix.

In the second optimization step for the sensitive mode, we need the gradient of $\mathcal{F} = \frac{\lambda}{2}\|\mathbf{A}_n''^\top\mathbf{A}_n'\|_{\mathrm{F}}^2$, which is calculated by $\frac{\partial\mathcal{F}}{\partial\mathbf{A}_n'} = \lambda\mathbf{A}_n''(\mathbf{A}_n'')^\top\mathbf{A}_n'$.

The entire optimization process is described in Algorithm 1. We can also use Newton's method to replace gradient descent, which has the advantages of fast convergence speed and less effort of tedious learning rate tuning. Newton's method requires the second-order derivative of $\mathcal{F}$, which is calculated by: $\frac{\partial^2\mathcal{F}}{\partial\mathbf{A}_n'\partial\mathbf{A}_n'^\top} = \lambda\mathbf{A}_n''\mathbf{A}_n''^\top$.

Finally, line 8 of Algorithm 1 should be modified to be "Update $\mathbf{A}_n' \leftarrow \mathbf{A}_n' - (\frac{\partial^2\mathcal{F}}{\partial\mathbf{A}_n'\partial\mathbf{A}_n'^\top})^\dagger\frac{\partial\mathcal{F}}{\partial\mathbf{A}_n'}$ ".

4 GENERALIZING FATR

So far, we have focused on a single binary sensitive attribute. We show here how to handle multi-feature cases (i.e., there are more than one sensitive attributes) and multi-category cases (i.e., the attribute can take more than two values). We also consider multisided attributes (i.e., more than one mode is considered sensitive), which is important in real-world applications [3]. Such multi-feature and multi-category cases are challenging for traditional regularization-based approaches [14, 29] since a regularization term can only account for fairness between two groups defined by one binary attribute. By missing the multi-way interactions among multiple categorical sensitive attributes, such a regularization-based approach may lead to less effective (and less fair) recommendation. However, the multi-feature and multi-category problems fit naturally into the proposed FATR framework.

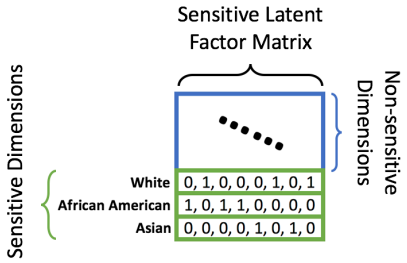


Figure 3: In the case of multi-category sensitive dimensions (e.g., by ethnicity), this example shows how to generate the sensitive latent factor matrix.

For the multi-feature case, we need to put all the sensitive features into the corresponding sensitive latent factor matrices, and add the orthogonal constraints to all the sensitive modes to isolate and extract all the sensitive information. For the multi-category case, we need to have c columns in the sensitive dimensions if the attribute can take c distinct values. Hence, the binary-feature case is just a special multi-category case where $c = 2$. Every dimension only indicates one specific category, for example, dimension i has value 1 for the entities who belong to category c_i and has value 0 for other instances. One example is shown in Figure 3.

For ease of presentation, we assume there are three sensitive attributes, one is denoted as S_1 belonging to the mode-n_1, another two are denoted as S_2 and S_3 belonging to the mode-n_2. And all of them have three available categories to take. For example, in the Twitter experts recommender, we want to enhance the fairness for experts with different genders (Female, Male, and Unspecified) and with different ethnicities (African-American, Asian, and White), and at the same time we also want to augment the fairness for the topics with different numbers of experts (small, medium, and large). The sensitive features of S_1 is $\mathbf{S}_1$ which has 3 columns. The sensitive features of S_2 and S_3 are $\mathbf{S}_2$ and $\mathbf{S}_3$, and concatenate them together to be $\mathbf{S}_{2,3}$ which has 6 columns. Then the objective function is:

$$\underset{\mathcal{X},\mathbf{A}_1\ldots\widetilde{\mathbf{A}_{n_1}}\ldots\widetilde{\mathbf{A}_{n_2}}\ldots\mathbf{A}_N}{\text{minimize}} \quad \mathcal{L} = \|\mathcal{X} - [\![\mathbf{A}_1\ldots\widetilde{\mathbf{A}_{n_1}}\ldots\widetilde{\mathbf{A}_{n_2}}\ldots\mathbf{A}_N]\!]\|_{\mathrm{F}}^2 + \frac{\lambda}{2}(\|\mathbf{A}_{n_1}''^\top\mathbf{A}_{n_1}'\|_{\mathrm{F}}^2 + \|\mathbf{A}_{n_2}''^\top\mathbf{A}_{n_2}'\|_{\mathrm{F}}^2) + \frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2$$

$$\text{subject to} \quad \Omega \circledast \mathcal{X} = \mathcal{T},$$
$$\widetilde{\mathbf{A}_{n_1}} = [\mathbf{A}_{n_1}' \ \ \mathbf{A}_{n_1}''],$$
$$\widetilde{\mathbf{A}_{n_2}} = [\mathbf{A}_{n_2}' \ \ \mathbf{A}_{n_2}''],$$
$$\mathbf{A}_{n_1}'' = \mathbf{S}_1, \ \ \mathbf{A}_{n_2}'' = \mathbf{S}_{2,3},$$

where $\widetilde{\mathbf{A}_{n_1}}$ and $\widetilde{\mathbf{A}_{n_2}}$ are the sensitive latent factor matrices, $\mathbf{A}_{n_1}'$ and $\mathbf{A}_{n_2}'$ are non-sensitive dimensions of $\widetilde{\mathbf{A}_{n_1}}$ and $\widetilde{\mathbf{A}_{n_2}}$, $\mathbf{A}_{n_1}''$ and $\mathbf{A}_{n_2}''$ are the sensitive dimensions of $\widetilde{\mathbf{A}_{n_1}}$ and $\widetilde{\mathbf{A}_{n_2}}$ which have the same values as $\mathbf{S}_1$ and $\mathbf{S}_{2,3}$.

We can still use Algorithm 1 to solve the new objective function with only line 8 and line 9 modified to update both $\widetilde{\mathbf{A}_{n_1}}$ and $\widetilde{\mathbf{A}_{n_2}}$. In the same way, the proposed method can be applied to model the cases with more sensitive features and more categories.

5 EXPERIMENTS

In this section, we empirically evaluate the proposed approach w.r.t three aspects – recommendation quality, recommendation fairness, and effectiveness of eliminating sensitive information – over four scenarios: (i) under the traditional matrix scenario; (ii) then by comparing matrix to tensor approaches; (iii) by varying the degrees of bias and sparsity to better explore their impact; and (iv) evaluating FATR's generalizability to the multi-feature and multi-category scenario.

5.1 Datasets

We consider a real-world movie dataset, a real-world social media dataset, and a collection of synthetic datasets for which we can vary degrees of bias and sparsity. We report the average results over three runs for all datasets.

MovieLens. We use the MovieLens 10k dataset [10], keeping all movies with at least 35 ratings. Following previous works [12, 15], we use the year of the movie as a sensitive attribute and consider movies before 1996 as old movies. Those more recent are considered new movies. In total, we have 671 users, 373 old movies, and 323 new movies. The sparsity of the dataset is 11.4%. Since we focus on implicit recommendation, we consider ratings to be 1 if the original ratings are higher than 3.5, otherwise 0. Then we have 15,579 positive ratings for new movies and 20,387 positive ratings for old movies, which reflects the bias in the dataset. We randomly split the dataset into 90% for training and 10% for testing.

User-Expert-Topic Twitter Data. We use a Twitter dataset introduced in [8] that has 589 users, 252 experts, and 10 topics (e.g., news, sports). There are 16, 867 links from users to experts across these topics capturing that a user is interested in a particular expert. The sparsity of this dataset is 1.136%. We consider race as a sensitive attribute and aim to divide experts into two groups: whites and non-whites. We apply the Face++ (https://www.faceplusplus.com/) API to the images of each expert in the dataset to derive ethnicity. In total, we find 126 whites and 126 non-whites, with 11,612 positive ratings for white experts but only 5,255 for non-whites. Since this implicit feedback scenario has no negative observations, we randomly pick unobserved data samples to be negative feedback with probability of 0.113% (one tenth of the sparsity). We randomly split the dataset into 70% training and 30% testing.

Synthetic Expert Datasets. To gauge the impact of degrees of bias and sparsity, we further generate a suite of synthetic expert datasets. We first generate three latent factor matrices by uniform distribution for user, expert, and topic, which are $\mathbf{U} \in \mathbb{R}^{200\times30}$, $\mathbf{E} \in \mathbb{R}^{100\times30}$, and $\mathbf{T} \in \mathbb{R}^{5\times30}$. Second, we set the last dimension of $\mathbf{E}$ to be the binary sensitive features to indicate two groups and make the numbers of the two groups equal. Third, we add constant values v_u and v_t to the sensitive dimensions of $\mathbf{U}$ and $\mathbf{T}$ to increase the bias. Then, we get the preference ratings tensor of size $200 \times 100 \times 5$ by calculating the Khatri-Rao product of $\mathbf{U}$, $\mathbf{E}$, and $\mathbf{T}$. Last, we set 1 to ratings lager than 0.5, meaning the user selects the expert with respect to the topic and set 0 to ratings less than 0.5, meaning the user does not select the expert with respect to the topic. We randomly sample the 1's based on a probability p to produce the final observed dataset. By adjusting the values of v_u and v_t, we generate datasets with varying imbalance of the proportion of the number of the positive ratings for the protected group over the total number of the positive ratings. With a proportion of 0.1, only 10% of positive ratings are for the protected group. We call this an extreme bias case. Similarly, we generate datasets with high bias (0.2), middle bias (0.3), and low bias (0.4). We further generate three levels of sparsity, which are 0.01 (high sparsity), 0.02 (middle sparsity), and 0.03 (low sparsity) by adjusting p. As a result, we have 12 different datasets: High Bias / High Sparsity, High Bias / Middle Sparsity, etc. All datasets are randomly split into 70% for training and 30% for testing.

5.2 Metrics

We consider metrics to capture recommendation quality, recommendation fairness, and the impact of eliminating sensitive information.

Recommendation Quality. To measure *recommendation quality*, we adopt **Precision@k** (P@K) and **Recall@k** (R@K), defined as:

$$P@k = \frac{1}{|\mathbb{U}|}\sum_{u\in\mathbb{U}}\frac{|\mathbb{O}_u^k \cap \mathbb{O}_u^+|}{k}, \quad R@k = \frac{1}{|\mathbb{U}|}\sum_{u\in\mathbb{U}}\frac{|\mathbb{O}_u^k \cap \mathbb{O}_u^+|}{\mathbb{O}_u^+},$$

where $\mathbb{O}_u^+$ is the set of items user u gives positive feedback to in test set and $\mathbb{O}_u^k$ is the predicted top-k recommended items. We also consider **F1@k** score, which can be calculated by $F1@k = 2 \cdot (P@k \times R@k)/(P@k + R@k)$. We set $k = 15$ in our experiments.

Recommendation Fairness. To measure *recommendation fairness*, we use two complementary metrics. The first one is the absolute difference between mean ratings of different groups (**MAD**):

$$MAD = |\frac{\sum R^{(0)}}{|R^{(0)}|} - \frac{\sum R^{(1)}}{|R^{(1)}|}|,$$

where $R^{(0)}$ and $R^{(1)}$ are the predicted ratings for the two groups and $|R^{(i)}|$ is the total number of ratings for group i. Larger values indicate greater differences between the groups, which we interpret as unfairness.

The second measure is the Kolmogorov-Smirnov statistic (**KS**), which is a nonparametric test for the equality of two distributions. The KS statistic is defined as the area difference between two empirical cumulative distributions of the predicted ratings for groups:

$$KS = |\sum_{i=1}^{T} l \times \frac{\mathfrak{G}(R^{(0)}, i)}{|R^{(0)}|} - \sum_{i=1}^{T} l \times \frac{\mathfrak{G}(R^{(1)}, i)}{|R^{(1)}|}|,$$

where T is the number of intervals for the empirical cumulative distribution, l is the size of each interval, $\mathfrak{G}(R^{(0)}, i)$ counts how many ratings are inside the i^{th} interval for group 0. In our experiments, we set $T = 50$. Lower values of KS indicate the distributions are more alike, which we interpret as being more fair.

MAD and KS can be directly applied to binary sensitive attributes. For multi-category cases, we need to calculate MAD and KS statistics for every dominant group vs. protected group pair among the

categories. For example, for the attribute of ethnicity with three categories – White (W), African-American (AA) and Asian (A), where AA and A are the two groups to be protected – we need to calculate the MAD and KS metrics for two pairs – W vs. AA, and W vs. A.

Note that we measure the fairness in terms of MAD and KS metrics across groups rather than within individuals, since absolute fairness for every individual may be overly strict and in opposition to personalization needs of real-world recommenders.

Eliminating Sensitive Information. To evaluate the impact of *eliminating sensitive information*, we use the sum of absolute cosine angles between non-sensitive and sensitive dimensions (**SCos**):

$$SCos = \sum_{i=1}^{r-2} \sum_{j=r-1}^{r} |cos(\mathbf{A}_i, \mathbf{A}_j)|,$$

where $\mathbf{A}_i$ and $\mathbf{A}_j$ are one non-sensitive dimension and one sensitive dimension indexed by i and j, and cos calculates the cosine angle between two vectors.

We also use the sum of absolute *Pearson correlation coefficient* between non-sensitive and sensitive dimensions (**SCorr**) to quantify the sensitive information:

$$SCorr = \sum_{i=1}^{r-2} \sum_{j=r-1}^{r} |corr(\mathbf{A}_i, \mathbf{A}_j)|,$$

where $corr$ calculates the *Pearson correlation coefficient* between two vectors. The lower the SCos and SCorr are, the better the sensitive information elimination result is.

For multi-category cases, Scos and Scorr should be calculated for every category separately to evaluate whether the impact of the multi-category attribute is eliminated with respect to all categories. Following our ethnicity example from earlier, we need to calculate SCos and SCorr for W, AA, and A separately.

5.3 Baselines

To evaluate the proposed FATR, we consider two variations – one using Gradient Descent (**FT(G)**) and one using Newton's Method (**FT(N)**) – in comparison with two tensor-based alternatives:

- *Ordinary Tensor Completion (**OTC**)*: The first is the conventional CP-based tensor completion method using ALS optimization algorithm as introduced in Section 2.2. This baseline incorporates no notion of fairness, so it will provide a good sense of the state-of-the-art recommendation quality we can achieve.
- *Regularization-based Tensor Completion (**RTC**)*: The second one is an extension from the fairness-enhanced matrix completion with regularization method introduced in [12, 14, 28], which adds a bias penalization term to the objective function. For tensor-based recommenders, we can use the regularized objective function (8) to enforce the statistical parity.

$$\begin{aligned} \underset{\mathcal{X}, \mathbf{A}_1, \ldots, \mathbf{A}_N}{\text{minimize}} \quad & \mathcal{L} = \|\mathcal{X} - [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\|_F^2 \\ & + \frac{\lambda}{2}\Big(\frac{1}{n_0}\|\mathbf{\Omega_0} \circledast [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\|_F^2 \\ & - \frac{1}{n_1}\|\mathbf{\Omega_1} \circledast [\![\mathbf{A}_1, \ldots, \mathbf{A}_N]\!]\|_F^2\Big)^2 \\ \text{subject to} \quad & \mathbf{\Omega} \circledast \mathcal{X} = \mathcal{T}, \end{aligned} \tag{8}$$

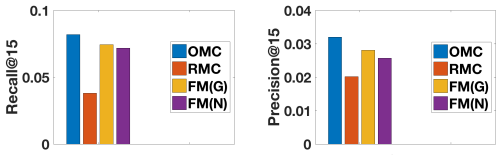


Figure 4: Recommendation quality (MovieLens).

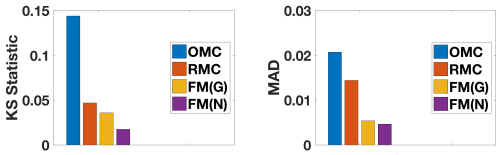


Figure 5: Recommendation fairness (MovieLens).

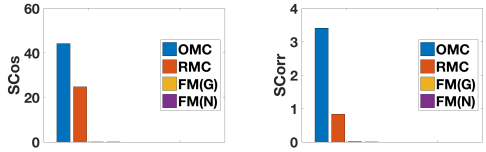


Figure 6: Eliminating Sensitive Information (MovieLens).

where $\lambda > 0$ is the regularization coefficient, $\mathbf{\Omega_0}$ and $\mathbf{\Omega_1}$ are the indicator tensors to indicate the ratings of the two groups determined by the binary sensitive attribute, n_0 and n_1 are the numbers of ratings to the two groups. We use Gradient Descent to solve this optimization problem.

Since the MovieLens data has only two modes (users and movies), we consider matrix versions of our tensor based methods (named **FM(G)** and **FM(N)**) versus matrix baselines of *Ordinary Matrix Completion (**OMC**)* and *Regularization-based Matrix Completion (**RMC**)* corresponding to RTC.

5.4 Matrix-Based Methods (MovieLens)

For the first experiment, we evaluate the four matrix-based approaches (OMC, RMC, FM(G) and FM(N)) over the MovieLens dataset. We set 50 as the latent dimension for all the methods and fine tune all other parameters; for our proposed methods we set $\lambda = 1$, $\gamma = 0.05$ and learning rate as 0.001 for FM(G), and $\lambda = 0.00001$ and $\gamma = 0.01$ for FM(N).

We begin by considering the quality of recommendation of the four approaches in Figure 4. As expected, the baseline with no notion of fairness – OMC – results in the best overall precision and recall. Of the three fairness-aware approaches, the regularization-based approach – RMC – performs considerably below the others, with our two approaches (FM) providing performance fairly close to OMC. This suggests that recommendation quality can be preserved, but leaves open the question of whether we can add fairness.

Hence, we turn to the impact on fairness of the four approaches. Figure 5 presents the KS statistic and MAD (recall, lower is better). We can see that all three fairness-aware approaches – RMC, FM(G) and FM(N) – have a strong impact on the KS statistic in comparison with OMC. And for MAD, we see that both FM(G) and FM(N) achieve much better ratings difference in comparison with RMC, indicating that we can induce aggregate statistics that are fair between the two sides of the sensitive attribute (old vs. new).

Last, we exam how well do these approaches perform from the perspective of sensitive information elimination. The left figure in Figure 6 shows the SCos statistic, while the right figure shows the

Methods	R@15	P@15	KS	MAD	SCos	SCorr
OMC	0.3467	0.0842	0.1660	0.0122	7.8035	1.9131
OTC	0.4384	0.0958	0.3662	0.0333	21.9193	8.7732
RMC	0.1609	0.0702	0.1521	0.0086	15.3268	0.8534
RTC	0.3003	0.0515	0.2003	0.0171	23.6818	1.4036
FM(G)	0.4045	0.0891	0.0523	0.0037	0.3081	0.1407
FT(G)	0.4180	0.0870	0.0195	0.0024	0.0936	0.0396
FM(N)	0.3298	0.0687	0.0245	0.0044	0.0022	0.0115
FT(N)	0.3975	0.0786	0.0173	0.0029	0.0001	0.0001

Table 2: Comparison for recommending Twitter experts.

SCorr statistic. Both of them demonstrate that the proposed FATR framework can eliminate sensitive information to a great extent, but RMC can only reduce the SCos to around half of that of OMC and SCorr to around one third of that of OMC.

5.5 Matrix vs. Tensor-Based Methods (Twitter)

We next turn to evaluating the expert recommendation task over the real-world Twitter dataset. Here we consider the tensor-based approaches – OTC, RTC, plus FT(G) and FT(N). To further evaluate the impact of moving from a matrix view to a tensor view, we also consider the purely matrix-based approaches, which compute users preferences on experts for each topic independently. We set 20 as the latent dimension for all the methods and fine tune all other parameters; for our proposed methods we set $\lambda = 1$, $\gamma = 0.05$ and learning rate as 0.001 for FM(G), and $\lambda = 0.00001$ and $\gamma = 0.01$ for FM(N). We show the results for all of our metrics in Table 2.

First, let's focus on the differences between matrix and tensor approaches. We observe that the tensor-based approaches mostly provide better recommendation quality (Precision@k and Recall@k) in comparison with the matrix-based approaches. Since the expert dataset is naturally multi-aspect, the tensor approaches better model the multi-way relationships among users, experts, and topics. We see that the fairness quality (KS and MAD) of matrix-based methods are better than tensor-based ones for the baselines methods (OMC vs OTC, and RMC vs RTC), but the fairness improves for our proposed methods when we move from matrix to tensor. We see a similar result for the impact on eliminating sensitive information (SCos and SCorr).

Second, let's consider the empirical results across approaches. We see that: (i) the proposed methods are slightly worse than OTC from the perspective of recommendation quality, but keep the difference small, and FM methods also have comparable recommendation performance with OMC; (ii) FT(G) and FT(N) provide the best fairness enhancement results, and FM(G) and FM(N) also alleviate the unfairness a lot compared with other matrix-based methods. RTC and RMC improve the fairness as well, but their effects are not as good as the proposed methods; (iii) the proposed approaches can effectively eliminate the sensitive information; and (iv) comparing the two variations of FATR, FT(G) always provides better recommendation quality but performs worse than FT(N) in terms of fairness enhancement and sensitive information elimination, which may be because Newton's method has stronger effects on optimization leading to more effective minimization of the orthogonal constraint term $\mathcal{F} = \frac{\lambda}{2}\|\mathbf{A}_n''^{\top}\mathbf{A}_n'\|_{\mathrm{F}}^2$ in Equation (3).

In addition, the $\frac{\gamma}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\|_{\mathrm{F}}^2$ term in our proposed objective function (3) may influence the recommendation performance, but the baselines do not have it, which may be an unfair comparison. Therefore, we do another experiment using OTC, RTC, OMC, and RMC with the L2-norm term. The recommendation performance results and fairness enhancement results are shown in Figure 7. We can conclude similarly that the proposed methods still perform well in terms of both recommendation quality and fairness enhancement. Besides, we find that compared with the baselines without L2-norm terms, the baselines with L2-norm have better recommendation quality but higher bias.

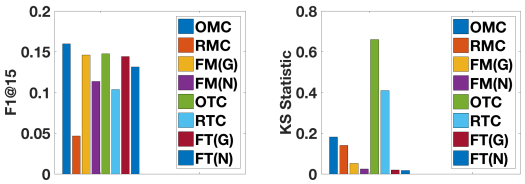
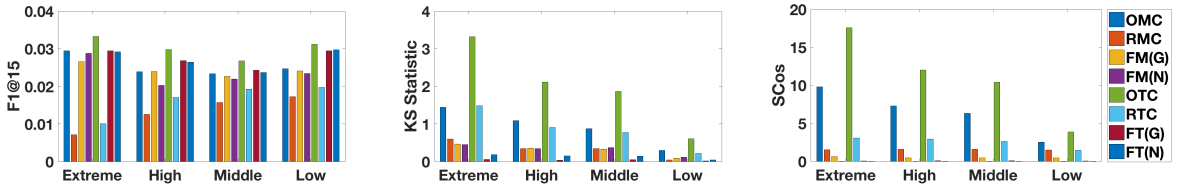


Figure 7: F1@15 and KS statistics of the proposed methods and the baselines with L2-norm terms.

5.6 Varying Bias and Sparsity (Synthetic)

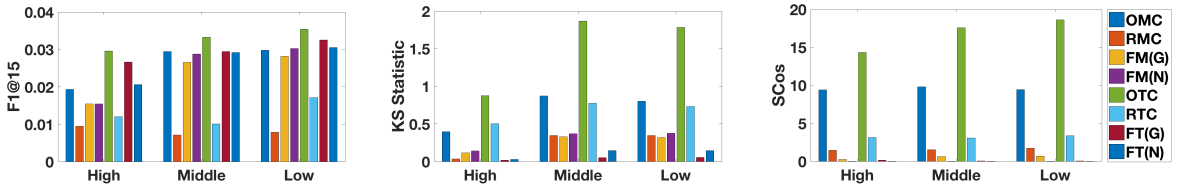
Next, we consider the impact of bias and sparsity through a series of experiments over the synthetic expert datasets. For parameters setting, the latent factor dimension is set as 20, we set $\lambda = 0.25$, $\gamma = 0.05$, learning rate as 0.002 for FT(G), and $\lambda = 0.0001$ and $\gamma = 0.1$ for FT(N). We set the latent dimension smaller than 30 on purpose, which is the number of factors we use when generating the synthetic dataset, because in practice, researchers tend to use low dimensional latent factor to model user-item interactions.

We begin by investigating the impact of bias – do our methods perform well even in cases of extreme bias? Or do they require only moderate amounts? We fix the sparsity level at 0.02 and vary the bias levels from Low, Middle, High, and Extreme. We show in Figure 8a the *F1@15* of all eight methods on these four datasets. The results show that OTC always performs best, but FT(G) does not reduce the *F1 score* much compared with other methods. Overall, tensor-based methods outperform matrix-based methods. And within matrix-based methods, FM(G) is just a little worse than OMC, and much better than RMC. Further, we can observe that as the bias level goes down, the recommendation quality is improved for all six fairness-aware methods in comparison with OTC and OMC. For example the F1@15 score difference between OTC and FT(G) are 0.0041, 0.0034, 0.0031, and 0.0015 for the extreme, high, medium, and low bias situations respectively. Figure 8b shows that for all the bias levels, the proposed FT(G) and FT(N) can enhance the fairness to a great extent. We can also observe that RTC and RMC can reduce the unfairness compared with conventional completion methods, but their performances are not comparable with the proposed methods. One outlier is the result produced by RMC in the low bias dataset. Although it reduces the KS as low as proposed methods do, its recommendation quality is not ideal. We also study how well do these methods eliminate sensitive information as demonstrated in Figure 8c. The figure shows that the proposed methods (both tensor-based and matrix-based) have the lowest SCos values, meaning that our methods can effectively eliminate the sensitive information. From these results, we can conclude that the proposed approaches provide good and consistent performance over all the bias levels.



(a) Recommendation quality: F1@15. (b) Fairness: KS Statistic. (c) Sensitive information elimination: SCos.

Figure 8: Evaluating the impact of bias (Synthetic Experts dataset).



(a) Recommendation quality: F1@15. (b) Fairness: KS Statistic. (c) Sensitive information elimination: SCos.

Figure 9: Evaluating the impact of sparsity under extreme bias (Synthetic Experts dataset).

Furthermore, we also analyze the results for datasets with various sparsities with bias level fixed at the extreme level. The results are shown in Figure 9. We can draw the similar conclusion from it that the proposed methods reduce the unfairness without much loss of the prediction accuracy for different sparsities. However, in addition to this conclusion, these results also imply that with the dataset being denser, the unfairness is more severe. Combining the observations from Figure 8 and Figure 9, we can learn that: (i) tensor completion possesses more algorithmic bias than matrix completion does; and (ii) the proposed FATR methods have consistent fairness-enhancement and sensitive information eliminating performance on datasets with various bias levels and sparsities. We also compute MAD and SCorr statistics, showing similar patterns as KS and SCos.

5.7 Multiple Features and Multiple Categories

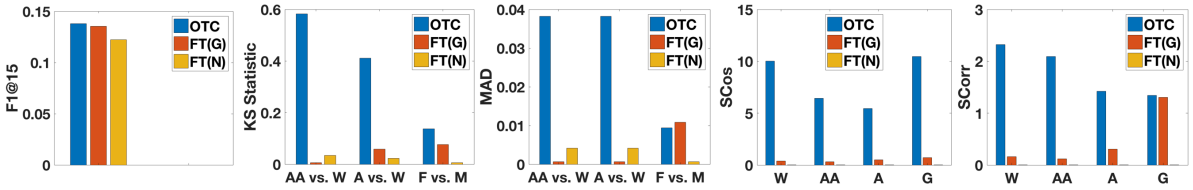
Finally, by the same dataset as used in Section 5.5, we investigate how the proposed model performs with multiple features and multiple categories (as introduced in Section 4). We consider both gender and ethnicity as sensitive attributes. For ease of experimentation, we consider gender (G) as a binary feature (M=Male, F=Female). For ethnicity, we consider three categories: White (W), African-American (AA), and Asian (A). Our dataset contains 126 whites with 11,612 positive feedbacks, 80 Asian people with 2,238 feedbacks, and 46 African-Americans with 3,017 positive feedbacks. The distribution of the gender is: 163 males and 83 females. Males have 10,160 positive ratings and females have 6,707 positive ratings. Other settings of the experiment are the same as single-feature experiment as described in Section 5.5.

For the parameters settings, we set the latent factor dimension as 20 for OTC, but 25 for FT(G) and FT(N) because there are 5 dimensions occupied by the sensitive dimensions, and we want similar degree of freedom for all the methods. We set $\lambda = 0.05$, $\gamma = 0.05$, and the learning rate 0.002 for FT(G), and $\lambda = \gamma = 1$ for FT(N). Because regularization-based models cannot be easily applied to this scenario, we compare FT(G) and FT(N) with OTC.

Figure 10a illustrates that the proposed methods can keep a relatively high recommendation quality compared with the OTC. Figure 10b shows that FT(N) model have a good fairness enhancement performance for both attributes. FT(G) works well on the ethnicity feature but a little unsatisfactory for the gender feature. One possible reason is that FT(G) requires more effort for parameter tuning. Moreover, the bias related to the ethnicity feature is more severe than the unfairness related to the gender feature, which makes it harder for the model to decrease the unfairness for the gender feature. Figure 10c shows the relationships between the latent factor matrices from the three methods and all the sensitive features. It implies that the FATR models can alleviate the impact of the sensitive information from all the sensitive attributes. Further, we see that FT(N) works well for all attributes including gender (which is challenging for the other approaches).

6 RELATED WORK

Friedman [7] defined that a computer system is biased "if it systematically and unfairly discriminates against certain individuals or groups of individuals in favor of others." As we have mentioned, considerable efforts have focused on classification tasks (e.g., recidivism prediction, loan approval) [5, 22, 25, 30, 31]. In the context of recommenders, Kamishima et al. first claimed the importance of neutrality in recommendation [13], and proposed two methods to enhance the fairness in explicit recommender systems. One is a regularization-based matrix completion method [14], another is a graph model-based method [15]. The performances of these two methods are similar. Later, Kamishima et al. extended the work in [14] to tackle the challenge of implicit recommendation problem [12]. Yao et al. [28, 29] proposed four novel metrics for fairness in collaborative filtering recommender systems and used similar regularization-based optimization approach as Kamishima did in [12, 14] to address the problems caused by different forms of bias. Moreover, there are some literatures working on fairness-enhancement for more specific scenarios. Abdollahpouri et al. [1] used a regularization-based matrix completion method to control popularity bias in learning-to-rank recommendation. Xiao et al. [27] proposed a multi-objective optimization model to implement fairness-aware group recommendation. Burke et al. [4] also



(a) Recommendation quality: F1@15. (b) Fairness Quality: KS and MAD. (c) Sensitive information elimination: SCos and SCorr.

Figure 10: Evaluating the generalizing ability to multi features and multi categories.

used a regularization-based matrix completion method to balance neighborhood fairness in collaborative recommendation.

7 CONCLUSION AND FUTURE WORK

This paper proposes a novel framework – FATR – to enhance the fairness for implicit recommender systems while maintaining recommendation quality. FATR effectively eliminates sensitive information and provides fair recommendation with respect to the sensitive attribute. Further, unlike previous efforts, the proposed model can also handle multi-feature and multi-category cases. Extensive experiments show the effectiveness of FATR compared with state-of-the-art alternatives. In our continuing work, we are interested in generalizing our framework to consider alternative notions of fairness beyond statistical parity. By extending our framework in this direction, we can provide a more customizable approach for defining and deploying fairness-aware methods. We are also interested in exploring how to incorporate real-valued features into the framework for recommenders with explicit ratings, and in running user studies on the perceived change of fairness for our methods.

ACKNOWLEDGMENTS

This work is, in part, supported by DARPA (#W911NF-16-1-0565) and NSF (#IIS-1841138). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 42–46.
[2] Dan Bernhardt, Stefan Krasa, and Mattias Polborn. 2008. Political polarization and the electoral effects of media bias. *Journal of Public Economics* 92, 5-6 (2008), 1092–1104.
[3] Robin Burke. 2017. Multisided Fairness for Recommendation. *arXiv preprint arXiv:1707.00093* (2017).
[4] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. 2017. Balanced Neighborhoods for Fairness-aware Collaborative Recommendation. In *FATREC Workshop on Responsible Recommendation Proceedings*. 5.
[5] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. *arXiv preprint arXiv:1701.08230* (2017).
[6] Ayman Farahat and Michael C Bailey. 2012. How effective is targeted advertising?. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 111–120.
[7] Batya Friedman and Helen Nissenbaum. 1996. Bias in computer systems. *ACM Transactions on Information Systems (TOIS)* 14, 3 (1996), 330–347.
[8] Hancheng Ge, James Caverlee, and Haokai Lu. 2016. TAPER: A Contextual Tensor-Based Approach for Personalized Expert Recommendation.. In *RecSys*. 261–268.
[9] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*. 3315–3323.
[10] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2016), 19.
[11] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.
[12] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on Recommendation Independence for a Find-Good-Items Task. In *FATREC Workshop on Responsible Recommendation Proceedings*.
[13] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Enhancement of the Neutrality in Recommendation.. In *Decisions@ RecSys*. 8–14.
[14] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation.. In *Decisions@ RecSys*. 1–8.
[15] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Issei Sato. 2016. Model-Based Approaches for Independence-Enhanced Recommendation. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 860–867.
[16] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 643–650.
[17] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 79–86.
[18] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.
[19] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.
[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[21] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 831–840.
[22] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 560–568.
[23] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 727–736.
[24] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 81–90.
[25] Chris Russell, Matt J Kusner, Joshua Loftus, and Ricardo Silva. 2017. When worlds collide: integrating different counterfactual assumptions in fairness. In *Advances in Neural Information Processing Systems*. 6417–6426.
[26] L. Sweeney. 2013. Discrimination in online ad delivery. *Queue* 11, 3 (2013), 10.
[27] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 107–115.
[28] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. *arXiv preprint arXiv:1705.08804* (2017).
[29] Sirui Yao and Bert Huang. 2017. New Fairness Metrics for Recommendation that Embrace Differences. *arXiv preprint arXiv:1706.09838* (2017).
[30] Muhammad Bilal Zafar, Isabel Valera, Manuel Rodriguez, Krishna Gummadi, and Adrian Weller. 2017. From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*. 228–238.
[31] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 325–333.