Data augmentation and refinement for recommender system: A semi-supervised approach using maximum margin matrix factorization

Shamal Shaikh[a], Venkateswara Rao Kagita[a], Vikas Kumar[b,*], Arun K Pujari[c]

[a]*National Institute of Technology, Warangal, India*
[b]*University of Delhi, New Delhi, India*
[c]*Mahindra University, Hyderabad, India*

Abstract

Collaborative filtering (CF) has become a popular method for developing recommender systems (RSs) where ratings of a user for new items are predicted based on her past preferences and available preference information of other users. Despite the popularity of CF-based methods, their performance is often greatly limited by the sparsity of observed entries. In this study, we explore the data augmentation and refinement aspects of Maximum Margin Matrix Factorization (MMMF), a widely accepted CF technique for rating predictions, which has not been investigated before. We exploit the inherent characteristics of CF algorithms to assess the confidence level of individual ratings and propose a semi-supervised approach for rating augmentation based on self-training. We hypothesize that any CF algorithm's predictions with low confidence are due to some deficiency in the training data and hence, the performance of the algorithm can be improved by adopting a systematic data augmentation strategy. We iteratively use some of the ratings predicted with high confidence to augment the training data and remove low-confidence entries through a refinement process. By repeating this process, the system learns to improve prediction accuracy. Our method is experimentally evaluated on several state-of-the-art CF algorithms and leads to informative rating augmentation, improving the performance of the baseline approaches.

Keywords: Recommender System, Matrix Factorization, Maximum Margin, Rating Augmentation, Rating Refinement

1. Introduction

A recommender system (RS) is a subset of information filtering systems designed to deal with information overload. These systems sift the users through the information space

*Corresponding author

Email addresses: ss911876@student.nitw.ac.in (Shamal Shaikh), venkat.kagita@nitw.ac.in (Venkateswara Rao Kagita), vikas007bca@gmail.com (Vikas Kumar), arun.pujari@mahindrauniversity.edu.in (Arun K Pujari)

and eliminate the need for manual filtering of their possible choices over the entire product space. Recently, RS has been widely and successfully applied in several different applications, including e-commerce (Hidasi et al., 2015), e-library (Wang et al., 2018), e-learning (Manouselis et al., 2011), tourism (Wang et al., 2017), job recommendation (Dave et al., 2018), and drug recommendation (He et al., 2018). RS attempts to analyze the feedback of the users and recommends a product/item to a user based on past feedback of the users about the product/item and the user's taste based on past purchase history. Any RS aims to construct a probable recommendation set of items tailored for individual users that matches their needs and taste. In the literature, several techniques have been proposed to generate a recommendation that can be categorized into content-based (CB), collaborative filtering (CF), and hybrid approaches (Bobadilla et al., 2013; Burke, 2002). The CB approach generates the recommendation based on the match between the user's profiles and the item's contents. In contrast, the CF approach uses users' past preferences for items that are available in the form of feedback, either explicit, such as ratings and reviews, or implicit, discovered based on the actions that the user performs concerning items. On the other hand, the hybrid approach combines different techniques of CF and CB approaches.

Among several approaches for Recommender systems (RSs), collaborative filtering (CF) has emerged as a fundamental paradigm (Guo et al., 2014; Adomavicius & Tuzhilin, 2005; Mobasher et al., 2003). Despite the popularity of CF-based methods, their performance is often greatly limited by the number of observed entries. To alleviate the problem of data sparsity, researchers have introduced semi-supervised learning. Semi-supervised learning is a crucial concept that focuses on exploiting knowledge from a large amount of unlabeled data and the small amount of available labeled data. Most existing CF methods adopt domain adaptation wherein knowledge learned from side information such as review texts and pictural data like posters (Yu et al., 2020; Zhao et al., 2016) is transferred to the main domain. However, exploiting and integrating side information requires consistent knowledge across auxiliary domains (Zhang et al., 2020; Lu et al., 2013). For example, the knowledge captured from book reviews is consistent with the knowledge in the movie domain, as both domains overlap in genre representations. However, it is inconsistent with the knowledge in the clothing domain (Liu et al., 2019). Thus, there is a need to implement a solution that can handle data sparsity without the involvement of side information or identification of compatible domains.

In this paper, we propose a self-training-based CF approach for rating augmentation that eases the burden of finding compatible domains for knowledge acquisition. In machine learning research, many self-training based have been proposed for the classification task, and the area of recommender system has been largely untouched with self-training based models (Gao et al., 2022; Wu et al., 2018; Nartey et al., 2019; Xie et al., 2020b). The idea is to leverage the learning process by utilizing both a small portion of label data and a large amount of unlabelled data (Krishnapuram et al., 2004). The recommendation task over ordinal rating preferences can also be visualized as a classification task (Rennie & Srebro, 2005; Kumar et al., 2017b,a). In (Kumar et al., 2017a), maximum margin matrix factorization (MMMF), a popular recommendation model that provides prediction with high accuracy, is visualized as an extension of a two-class classifier to a unified multi-class classifier and proposed a method for ordinal matrix completion by hierarchically arranging multiple binary MMMF. This paper advances the frontier of research on this subject by investigating the data augmentation and refinement aspects which are not attempted so far by the researchers. The inherent characteristics of MMMF are exploited to assess the confidence level of algorithm's prediction of individual rating. We hypothesize that the any CF algorithm's predictions with low confidence can be due to some deficiency in the training data and hence, the performance of the algorithm can be improved by adopting a systematic data augmentation strategy. In the present study, we consider Maximum Margin Matrix Factorization as the base CF-based algorithm. We propose a self-training-based semi-supervised approach for rating augmentation to improve the performance of MMMF. The generic principle that we adopt here is to assess the algorithm's prediction confidence qualitatively as low or high. Our method is an iterative process in which the ratings of unknown items predicted with high confidence are used to augment the training data. At this stage, the refinement process removes an entry in the training data that is found to be of low confidence. By repeated application of the process of augmentation and refinement, the system learns to improve prediction accuracy. We experimentally evaluate the proposed rating augmentation technique by considering several state-of-the CF algorithms. The experimental result corroborates our claims that the proposed strategy leads to informative rating augmentation and improves the performance of the baseline approaches under consideration.

The rest of the paper is organized as follows. Section 2 briefly reviews existing approaches to handle data sparsity issues in the recommendation system. Section 3 summarizes the well-known existing MMMF process. We present the proposed ST-MMMF model of rating augmentation in Section 4. An experimental analysis of the proposed algorithm is reported in Section 5. Finally, Section 6 concludes and indicates several issues for future work.

2. Related Work

The traditional approaches to alleviating the problem of data sparsity can be broadly divided into two categories - methods that impute ratings to generate pseudo-rating data based on some estimation criterion and methods that use auxiliary information (Kuo et al., 2021; Natarajan et al., 2020; Duan et al., 2022). In this section, we present a brief overview of the significant approaches proposed to overcome the problem of data sparsity in the user-item rating matrix.

Rating imputation is a non-trivial task and requires sophisticated methods to infer the missing rating. Several approaches have been proposed in recent years for rating imputation, among which the mean-based imputation is the most naive. The rating for any unobserved (user, item)-pair can be imputed either with the user-specific mean or item-specific mean (Breese et al., 2013; Ghazanfar & Prugel, 2013). The mean method of rating imputation suffers from high bias because there are going to be so many entries in the rating matrix with a similar rating level which may lead to an imbalance rating distribution. Matrix factorization techniques, such as Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), have been widely used for imputing missing ratings in recommender systems (Kim & Yum, 2005; Ranjbar et al., 2015). These methods factorize the rating matrix into two low-rank matrices and use the factorized matrices to estimate the missing ratings. In (Hwang et al., 2016), SVD is applied on the pre-use preferences matrix to identify the uninteresting items and then assigns zero ratings. Collaborative filtering methods, such as user- and item-based, have also been used to rating imputation (Bell et al., 2007; Ghazanfar & Prugel, 2013). These methods use the similarity between users or items to estimate missing ratings (Ren et al., 2012). Bayesian methods, such as Probabilistic Matrix Factorization (PMF) and Bayesian Personalized Ranking (BPR), have been used to incorporate uncertainty in the imputation process (Mnih & Salakhutdinov, 2007; Rendle et al., 2012; Wang et al., 2019; Moon et al., 2023). These methods model the rating matrix as a probabilistic generative model and use Bayesian inference to estimate missing ratings. The other category of methods uses side information, such as reviews, images, and videos, to handle the data sparsity problem (Guo et al., 2019). Niu et al. (Niu et al., 2016) proposed a neighborhood-based approach of collaborative filtering where the reviews are used as side information to compute the similarity between two users. Ning et al. (Ning & Karypis, 2012) propose several methods to utilize the item side information to learn sparse linear coefficient matrix to the top-N recommendation. In (Strub et al., 2016), a Neural Network architecture called CFN to learn the non-linear representation of users and items. The authors introduce a novel loss function adapted to input data with missing values and utilize the side information. A deep hybrid model is proposed to learn deep users' and items' latent factors from the rating matrix and side information (Dong et al., 2017). In (Massa & Avesani, 2004), a trust matrix and the user-item rating matrix are used together to improve the accuracy of similarity calculation and recommendation quality.

3. Maximum Margin Matrix Factorization

Maximum Margin Matrix Factorization (MMMF) is a well-known method for computing a dense approximation $X \in \mathbb{R}^{N\times M}$ of a sparse matrix $Y \in \mathbb{R}^{N\times M}$ with ordinal entries (Srebro et al., 2004). The initial proposal for MMMF is formulated as a semi-definite programming (SDP) problem that can handle the factorization of small matrices. Rennie et al. (Rennie & Srebro, 2005) proposed a gradient-based optimization method for MMMF to factorize a matrix with sufficiently large size. We will refer to this as MMMF in our subsequent discussion. The readers are requested to refer (Rennie & Srebro, 2005; Kumar, 2019) for details.

3.1. MMMF as Gradient-based Optimization

The MMMF approach is primarily designed for collaborative filtering tasks where the user's preferences over items are observed in the form of ratings and are naturally organized in matrix form. Let $Y = [y_{ij}] \in \mathbb{R}^{N\times M}$ denote the rating matrix, where N represents the number of users, and M is the number of items. The entries y_{ij} are from $\{0, 1, 2, \ldots, R\}$, where R denotes the maximum rating level, and 0 denotes the unobserved entries. Given a sparse rating matrix Y, MMMF seeks a minimum trace norm matrix $X \in \mathbb{R}^{N\times M}$ that approximates the observed entries in matrix Y. The gradient-based optimization problem with a term $\|X\|_\Sigma$, trace norm of matrix X, is a complicated non-differentiable function for which it is difficult to find the subdifferential (Rennie & Srebro, 2005). Hence, instead of searching over X, MMMF learns a pair of low norm factor matrices $U \in \mathbb{R}^{N\times d}$ and $V \in \mathbb{R}^{M\times d}$ such that

$$Y \approx UV^T, \tag{1}$$

where d is a parameter.

The factor matrices U and V are obtained by minimizing the regularized loss function where the Froebenius norm of the factor matrices is used to upper bound the minimization objective with the term $\|X\|_\Sigma$, i.e., for any U and V, $\|UV^T\|_\Sigma \leq \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2)$. In order to map the predicted value $x_{ij} = U_i V_j^T$ into R intervals, MMMF also learns $R-1$ thresholds, $\theta_{i,1} \leq \theta_{i,2}, \ldots, \leq \theta_{i,R-1}$, for each ith user. The objective function of MMMF for ordinal rating predictions can be written as follows.

$$\min_{U,V} J(U,V,\Theta) = \sum_{y_{ij}|ij\in\Omega} \Big(\sum_{r=1}^{y_{ij}-1} h(U_i V_j^T - \theta_{i,r}) + \sum_{r=y_{ij}}^{R-1} h(\theta_{i,r} - U_i V_j^T)\Big) + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2), \tag{2}$$

where $\|\cdot\|_F$ is Frobenius norm, U_i and V_j denotes the ith row of matrix U and matrix V, respectively, $\Theta = [\theta_{ij}]$ is a threshold matrix consisting of $R-1$ thresholds for each ith user, $\lambda > 0$ is the regularization parameter, Ω is the set of observed entries and $h(z)$ is the smooth hinge loss defined as

$$h(z) = \begin{cases} 0, & \text{if z} \geq 1; \\ \frac{1}{2}(1-z)^2, & \text{if } 0 < z < 1; \\ \frac{1}{2} - z, & \text{otherwise.} \end{cases} \tag{3}$$

The equation given in (2) can be rewritten as follows.

$$\min_{U,V} J(U,V,\Theta) = \sum_{r=1}^{R-1} \sum_{(i,j)\in\Omega} h\big(T_{ij}^r(\theta_{i,r} - U_i V_j^T)\big) + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \tag{4}$$

where T is defined as

$$T_{ij}^r = \begin{cases} +1, & \text{if r} \geq y_{ij}; \\ -1, & \text{if r} < y_{ij}. \end{cases}$$

Several approaches can be used to optimize the objective function 4. A gradient descent method and its variants start with random U, V and Θ and iteratively update them using equations 5, 6 and 7, respectively.

$$U_{ip}^{t+1} = U_{ip}^t - c\frac{\partial J}{\partial U_{ip}^t}. \tag{5}$$

$$V_{jq}^{t+1} = V_{jq}^{t} - c\frac{\partial J}{\partial V_{jq}^{t}}. \tag{6}$$

$$\theta_{i,r}^{t+1} = \theta_{i,r}^{t} - c\frac{\partial J}{\partial \theta_{i,r}^{t}}. \tag{7}$$

c is the learning rate in the above equations and suffixes t and $(t+1)$ indicate current and updated values. The gradients of the variables to be optimized are determined as follows.

$$\frac{\partial J}{\partial U_{ip}} = \lambda U_{ip} - \sum_{r=1}^{R-1} \sum_{j|(i,j)\in\Omega} T_{ij}^{r} h'\big(T_{ij}^{r}(\theta_{i,r} - U_i V_j^T)\big) V_{jp} \tag{8}$$

$$\frac{\partial J}{\partial V_{jq}} = \lambda V_{jq} - \sum_{r=1}^{R-1} \sum_{i|(i,j)\in\Omega} T_{ij}^{r} h'\big(T_{ij}^{r}(\theta_{i,r} - U_i V_j^T)\big) U_{iq} \tag{9}$$

$$\frac{\partial J}{\partial \theta_{i,r}} = \sum_{j|(i,j)\in\Omega} T_{ij}^{r} h'\big(T_{ij}^{r}(\theta_{i,r} - U_i V_j^T)\big) \tag{10}$$

where $h'(z)$ is defined as follows.

$$h'(z) = \begin{cases} 0, & \text{if z} \geq 1; \\ z-1, & \text{if } 0 < z < 1; \\ -1, & \text{otherwise.} \end{cases} \tag{11}$$

Once U, V and θ's are computed, the matrix completion process is accomplished as follows.

$$\hat{y}_{ij} = \begin{cases} r, & \text{if } (i,j) \notin \Omega \wedge (\theta_{i,r} \leq x_{ij} \leq \theta_{i,r+1}) \wedge (0 \leq r \leq R-1); \\ y_{ij}, & \text{if } (i,j) \in \Omega, \end{cases}$$

where, $\hat{y}_{ij}$ is the prediction for item j by user i. For simplicity of notation, we assume $\theta_{i,0} = -\infty$ and $\theta_{i,R} = +\infty$ for each user i.

3.2. Geometrical Interpretation of MMMF

MMMF decomposes a large user-item rating matrix Y into two low-norm matrices U and V, the former representing users and the latter representing items. Geometrically, each row of user latent factor matrix U defines a hyperplane in the d-dimensional space, and similarly, the rows of V can be viewed as points in the same space. The objective is to learn from a sparse Y the decision hyperplane for each user to separate points of diferent ratings with the largest possible margin. When Y is bi-valued ($\{-1, +1\}$), the objective is to learn the hyperplane defining for each user (U_i for user i) a decision function that separates one rating ($+1$) from the other (-1) with maximum margin. In the case of ordinal scale, i.e., $y_{ij} \in \{1, 2, \ldots R\}$, The objective is to partition the set of items embedded as points into R-regions, each corresponding to a rating level, separated by parallal hyperplanes defined by U_i for user i. This is achieved by optimizing a smooth version of the hinge loss function well-known for margin maximization (Rennie & Srebro, 2005; Kumar et al., 2017a,b; Salman et al., 2016). The outcome of the optimization is to get the optimal combination of (U, V, Θ) such that the embedding of items rated as r fall as correctly as possible into regions defined by $(U_i, \theta_{i,r-1})$ and $(U_i, \theta_{i,r})$ with sufficient margin. For the simplicity of notation, we assume $\theta_{i,0} = -\infty$ and $\theta_{i,R} = +\infty$, for each ith user. Figure 1 illustrate this concept by taking a row of U as the decision hyperplane for a user, and rows of V are embedding of points corresponding to the items.

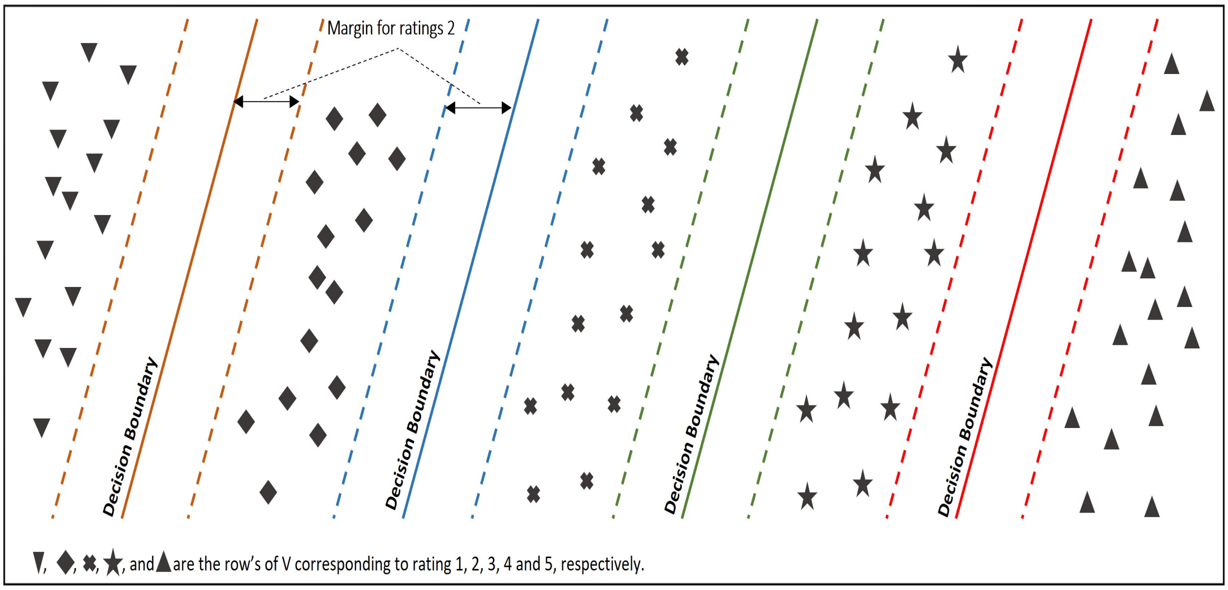


Figure 1: Classification by MMMF for the ith user

4. Proposed Algorithm

In this section, we describe the proposed data augmentation and refinement strategy, denoted as Self-Training with Maximum Margin Matrix Factorization (ST-MMMF), which is trained in a semi-supervised fashion to generate new rating samples. Data augmentation in ML allows artificially increasing the size of the training set by adding new synthetic examples and helps the decision function to become invariant to the changes (Xie et al., 2020a; Ratner et al., 2017). This can be interpreted as a regularization method that induces a useful bias by preventing the model from focusing on irrelevant features and making it less prone to overfitting (Chen et al., 2020). *Data augmentation* in the present context implies that some synthetic user-item ratings are introduced where such data is not available in the original user-item interaction record. In other words, it is to introduce some values when y_{ij} is unknown in Y. The choice of y_{ij} cannot be arbitrary, and hence, data augmentation is the judicious process of selecting unknown y_{ij}'s, which can be used to augment the sparse matrix Y. We show later that our augmentation process satisfies the desirable properties required for the purpose. The process of *data refinement* is yet another strategy adopted in ML research to improve the system's performance. In the present context, we design the proposed data refinement strategy by identifying certain weak y_{ij}'s known in the original training Y and remove them to refine the training set.

ST-MMMF follows an iterative process for *data augmentation* and *data refinement*. Each iteration of ST-MMMF is comprised of four pivotal stages, (1) Learning the latent factor matrices U, V, and Θ for the current Y, where U (V) represents the user (item) latent factor matrix, and Θ signifies the threshold matrix; (2) Augmenting Y with unobserved entries predicted with high confidence in Stage (1); (3) Refining the augmented Y by removing the known ratings indicated in Stage (1) as of low confidence; (4) Replacing the matrix Y with the refined and augmented matrix obtained after Stage (3) for the subsequent training iteration.

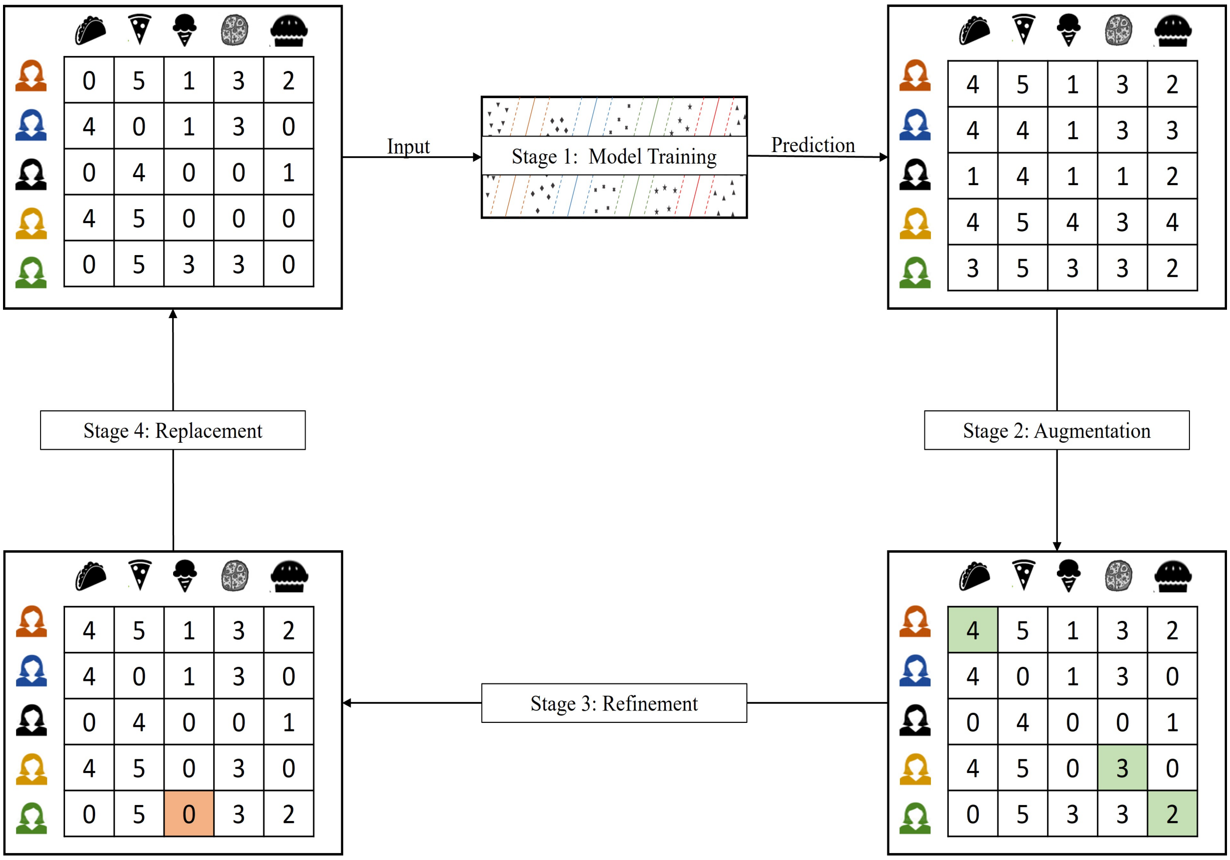


Figure 2: Overview of the ith iteration of ST-MMMF.

Figure 2 depicts a high-level overview of the proposed ST-MMMF approach showing various stages involved in an ith iteration of the proposed ST-MMMF. The matrix at the top left corner denotes the input matrix Y to ith iteration. In Stage 1, MMMF is employed to learn two latent factor matrices, U and V, representing users and items. Additionally, $R - 1$ thresholds are also learned for each user to map the corresponding real-valued predictions to the discrete rating scale. The matrix at the top right corner is the predicted matrix obtained after the application of the MMMF algorithm over the input matrix Y. Following the geometrical interpretation of MMMF, we have identified the highly confident predictions corresponding to the unobserved entries in the input matrix Y. Similarly, we have also marked the observed entries predicted with low confidence. The detailed procedure is explained in Section 4.1. In Stage 2, a subset of ratings identified to be predicted with high confidence is included in the matrix Y. The matrix at the bottom right corner shows the resultant matrix after the augmentation phase, with color-coded entries denoting the newly added ratings. Finally, in the refinement phase, the observed entries of matrix Y marked to be predicted with low confidence in Stage 1 are removed. The matrix shown in the bottom left corner, obtained after Stage 3, is given as input to the next iteration of ST-MMMF.

4.1. Data Augmentation and Refinement

In the original proposal of MMMF, the authors used the all-threshold hinge function to calculate the prediction loss corresponding to the observed entries in the rating matrix. Geometrically, the all-threshold hinge function not only tries to embed the points rated as r by the ith user into the region defined by $(U_i, \theta_{i,r-1})$ and $(U_i, \theta_{i,r})$ but also consider the position of the points with respect to other hyperplanes. This is more reasonable in discrete ordinal rating prediction, where it is always better to predict '2' than '5' if the true label is '1', unlike the multi-class classification setting where all mistakes are equal. Hence, it is desirable that V_j, an embedding for jth item rated by the ith user, should satisfy the condition $U_iV_j^T - \theta_{i,r-1} > 0$ for $r < y_{ij}$ and $U_iV_j^T - \theta_{i,r} < 0$ for $r \geq y_{ij}$. For every ith user, the points falling in the region defined by $(U_i, \theta_{i,r-1})$ and $(U_i, \theta_{i,r})$ are assigned rating r in the prediction.

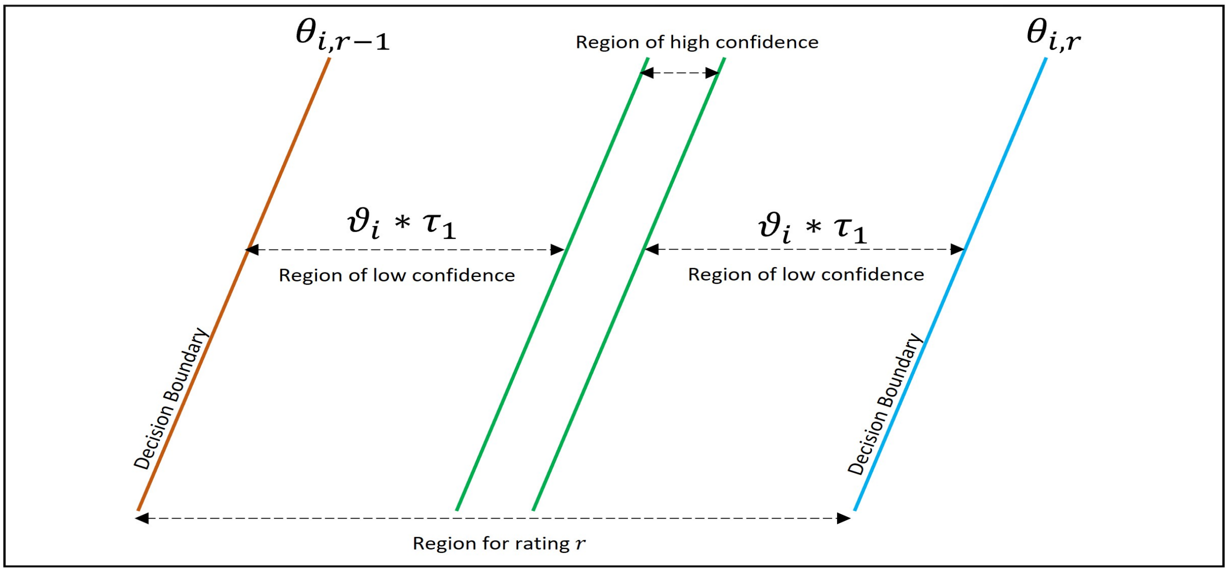


Figure 3: Overview of ST-MMMF for rating imputation.

ST-MMMF data augmentation and refinement process exploits the geometrical interpretation of MMMF. We assume that the region towards the center defined by $(U_i, \theta_{i,r-1})$ and

---

Algorithm 1: ST-MMMF (Y, d, K, λ, τ_1, τ_2)

Input : Rating Matrix: Y, Size of Latent Dimension Space: d, Regularization Parameter: λ_1, Parameters for Rating Augmentation and Refinement: τ_1 and τ_2, Sampling Percentage $= s$

Output: Predicted Rating Matrix: $\hat{Y}$

repeat
- $\Omega \leftarrow Index(Y \neq 0)$
- $Y^A = [0]^{N \times M}$
- $U, V, \Theta \leftarrow MMMF(Y, d, \lambda)$ /* $U \in \mathbb{R}^{N \times d}, V \in \mathbb{R}^{M \times d}$ and $\Theta = [\theta_{ij}] \in \mathbb{R}^{N \times R-1}$ */
- $X \leftarrow UV^T$
- $\hat{Y} \leftarrow$ Discretize (X, Θ)
- **for** $i \leftarrow 1$ **to** N **do**
  - $\vartheta_i \leftarrow$ AvgDist$(\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,R-1})$
- **end**
- /* Augmentation */
- **foreach** $ij \notin \Omega$ **do**
  - **for** $r \leftarrow 1$ **to** R **do**
    - /* $\theta_{i,0} \leftarrow -\infty$ and $\theta_{i,R} \leftarrow +\infty$ */
    - **if** *find 'r' satisfying Equation (12)* **then** $y^A_{ij} \leftarrow r$;
  - **end**
- **end**
- /* Refinement */
- **foreach** $ij \in \Omega$ **do**
  - **for** $r \leftarrow 1$ **to** R **do**
    - **if** *find 'r' satisfying Equation (13)* **then** $y_{ij} \leftarrow 0$;
  - **end**
- **end**
- $Y \leftarrow Y +$ Sample(Y^A, s)

until *stop criterion reached*;

$(U_i, \theta_{i,r})$ contains the points for which rating r is predicted with high confidence. Hence, the unobserved entry y_{ij} satisfying the following condition is augmented with rating r for the next round of the training process.

$$\theta_{i,r-1} + \vartheta_i * \tau_1 < U_i V_j^T < \theta_{i,r} - \vartheta_i * \tau_1 \tag{12}$$

Here, $\vartheta_i = \frac{1}{R-2}\sum_{r=2}^{R-1}(\theta_{i,r} - \theta_{i,r-1})$ is the average gap between consecutive pairs of threshold values learnt for the ith user. $\tau_1 < 0.5$ is a shifting parameter used to define the region towards the center bounded by $(U_i, \theta_{i,r-1})$ and $(U_i, \theta_{i,r})$. In other words, Equation (12) defines the region of high confidence for the ith user. Any unobserved entry y_{ij} satisfying Equation (12) is then correctly classified as rating r with high confidence. The high-confidence region for the rating augmentation is depicted in Figure 3. Similarly, the known y_{ij} falling closer the boundary defined by $(U_i, \theta_{i,r})$ can be termed as input data with low confidence. Hence, the observed entry y_{ij} satisfying the following condition is discarded for the next round of the training process.

$$\theta_{i,r} - \vartheta_i * \tau_2 < U_i V_j^T < \theta_{i,r} + \vartheta_i * \tau_2 \tag{13}$$

Here, ϑ_i is the same as defined previously, and $\tau_2 < \tau_1$ is a shifting parameter. Algorithm 1 outlines the main flow of the proposed method where the training set is augmented by adding all the ratings selected with (τ_1, s).

4.2. Handling Skewed Dataset

The ratings are nonuniformly distributed in many real-world datasets and therefore, in the data augmentation process, the rating label with the highest proportion tends to be augmented more frequently than that with a smaller proportion. This will result in the eventual degradation of the quality of the training set. We take into account rating distribution in the current training set while performing the augmentation to overcome the possible degradation. The frequency of augmenting rating labels with low distribution is made more frequently than those with high distribution. Let $\mathbb{Z}_i$ denote the ratio of ith label rating in the training set, and $S(\tau_1, s)$ is the set of ratings selected with the hyperparameters (τ_1, s) for the augmentation. We compute the total number of samples to be augmented to each rating label from the set $S(\tau_1, s)$ on the following basis.

$$\frac{(1-\mathbb{Z}_1)}{\sum_{j=1}^{R}(1-\mathbb{Z}_j)} : \frac{(1-\mathbb{Z}_2)}{\sum_{j=1}^{R}(1-\mathbb{Z}_j)} : \cdots : \frac{(1-\mathbb{Z}_R)}{\sum_{j=1}^{R}(1-\mathbb{Z}_j)} \tag{14}$$

Furthermore, to avoid the drastic change in rating distribution, the training set can be augmented with a small proportion of ratings predicted with high confidence. The maximum number of ratings that can be augmented in each iteration of ST-MMMF is set to 5000 for our experimental analysis.

4.3. Complexity Analysis

In this section, we analyze the computational complexity of the proposed method. The time complexity of ST-MMMF mainly comprises of two components: 1) Learning the latent factor matrices U and V corresponding to users and items, respectively, and a threshold matrix Θ consisting of $R-1$ thresholds for each user; and 2) Computation requires for augmentation and refinement. The optimization problem of MMMF given in Equation 2 requires major computations for matrix multiplication. For the simplicity of representation, we assume that the cost of multiplying two matrices of size $N \times d$ and $M \times d$ is $O(NMd)$.

The updation of matrix U requires $(R-1)$ time multiplication of two matrices of size $N \times M$ and $M \times d$ in the worst case. A similar computation is required for the updation of matrices V and Θ. Hence, the overall computation required to learn the matrices U, V, and Θ using MMMF is $O(3t_1(R-1)(NMd))$, that is, $O(t_1RNMd)$, where t_1 is the maximum number of gradient iteration (Veeramachaneni et al., 2019). The augmentation and refinement process inherently calculates two thresholds for each rating level defining the left and right boundary. During the augmentation phase, only unobserved entries are considered whereas the refinement phase takes into account observed entries. In summary, each real-valued prediction is compared with $(2 \times R)$ thresholds. Hence, the computation cost required for the augmentation and refinement phase is $O(2RMN)$, that is, $O(RNM)$. Let t_2 denote the maximum number of iterations required for ST-MMMF. Hence, the overall computation cost of the proposed method is $O(t_2(t_1RNMd + RNM))$, that is, $O(t_1t_2RNMd)$.

5. Experiments

This section reports the experimental analysis of the proposed ST-MMMF algorithm. The performance of the rating imputation task is evaluated in terms of the predictive performance of the proposed approach as compared to several existing baseline approaches.

Table 1: Characteristics of MovieLens dataset

Dataset	Users	Movies	Ratings	Scale	Sparsity
Movielens 100K	943	1682	$100,000$	1-5	94%
Movielens 1M	6040	3952	$1,000,209$	1-5	96%

5.1. Experimental Settings

We conduct experiments on two publicly available movie rating datasets: MovieLens 100K and MovieLens 1M [1]. The datasets are preprocessed, and users with less than 20 observed ratings are removed. The detailed characteristics of these datasets are reported in Table 1. Figure 5a and 5b show the distribution of ratings in MovieLens 100K and MovieLens 1M datasets, respectively.

We adopted the following two evaluation metrics most widely used to evaluate the recommendation performance of the methods, i.e., mean absolute error (MAE) and root mean square error (RMSE).

$$MAE = \frac{\sum_{ij\in\Omega} |y_{ij} - \hat{y}_{ij}}{|}|\Omega|, \quad RMSE = \sqrt{\frac{\sum_{ij\in\Omega}(y_{ij} - \hat{y}_{ij})^2}{|\Omega|}},$$

We consider SVD (Koren et al., 2009), NMF (Luo et al., 2014), SVD++ (Koren, 2008) and Co-Clustering (George & Merugu, 2005) as our baseline algorithms for the performance evaluation. We have used the Surprise (Hug, 2020) library to run the baseline

[1]`https://grouplens.org/datasets/movielens/`

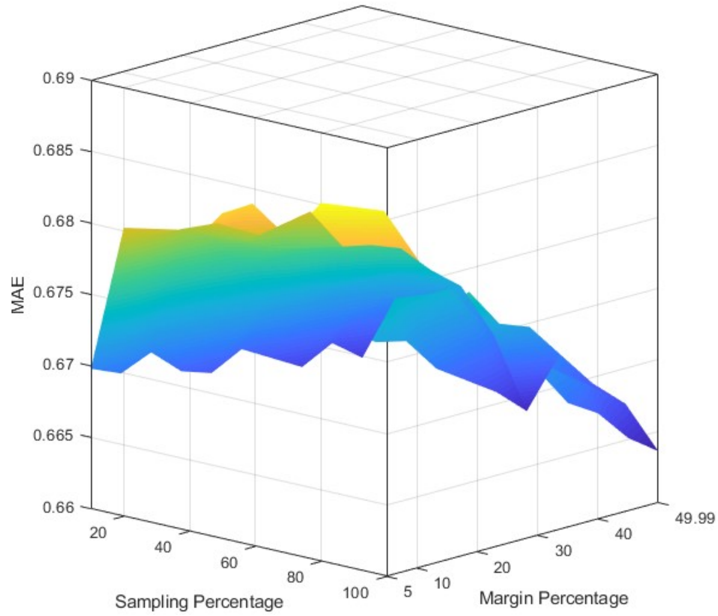


Figure 4: Influence of hyper-parameters τ_1 and s

algorithms and tunning the hyperparameters after each stage of the augmentation process. For ST-MMMF, the regularization value λ is tuned from the candidate set $\{10^{\frac{i}{16}}\}$, $\forall i \in \{1, 5, ..., 40\}$. The value of sampling percentage s in every iteration of ST-MMMF is searched in $\{10, 20, \ldots, 100\}$. The parameter τ_1 used to determine the high confidence region is tuned from $\{5, 10, 15, \ldots, 45, 49.99\}$ and the parameter τ_2 is fixed to 10, defining it as the low confidence region and refined noise in each iteration. Figure 4 shows the performance of ST-MMMF in terms of MAE on the Movielens 100K dataset. The MAE reported is an average of 50 runs for every pair of (τ_1, s) for a fixed value of regularization parameter λ. We select $\tau_1 = 49.99$ and $s = 100$ corresponding to the smallest MAE for subsequent experiments on both datasets. Furthermore, we randomly selected 80% of the observed ratings for training and the remaining 20% as the test set. The prediction accuracy averaged over three runs is reported.

5.2. Results and Analysis

We first analyze the effect of rating distributions on the prediction accuracy of each rating. It can be seen in 5a and 5b that the ratings exhibit imbalanced distribution, and the difference between the items rated 1 and 4 is significantly large. This would result in predictive bias where ratings with large samples may subsume ratings with a few samples (Kumar et al., 2017a). Our augmentation process discussed in Section 4.2 takes care of such imbalance rating distribution. To validate the proposed augmentation strategy, we have conducted some initial investigation using MMMF (Rennie & Srebro, 2005) over both datasets.

Table 2 and 3 show the effect of augmentation on individual rating prediction using MovieLens 100K and MovieLens 1M datasets, respectively. We also calculated label-wise

Table 2: Effect of imputation on MovieLens 100K dataset. '*' indicates that the measure is not applicable.

(a) Confusion matrix after the 1st iteration.

Actual	Predicted									
	1	2	3	4	5	$HR@0$	$HR@1$	$HR@2$	$HR@3$	$HR@4$
Training-set										
1	2086	1865	864	70	3	0.4268	0.3815	0.1768	0.0143	0.0006
2	160	3198	5296	432	10	0.3516	0.5998	0.0475	0.0011	*
3	9	225	16254	5173	55	0.7485	0.2486	0.0029	*	*
4	0	10	2809	23892	628	0.8739	0.1257	0.0004	0	*
5	0	4	159	7030	9767	0.5759	0.4145	0.0094	0.0002	0
Test-set										
1	231	248	505	230	8	0.1890	0.2029	0.4133	0.1882	0.0065
2	82	331	1263	583	15	0.1456	0.5915	0.2564	0.0066	*
3	30	285	2843	2146	125	0.5237	0.4478	0.0286	*	*
4	19	68	2033	4249	466	0.6217	0.3656	0.0099	0.0028	*
5	4	18	391	2692	1136	0.2679	0.6348	0.0922	0.0042	0.0009

(b) Confusion matrix after the 50th iteration.

Actual	Predicted									
	1	2	3	4	5	$HR@0$	$HR@1$	$HR@2$	$HR@3$	$HR@4$
Training-set										
1	26470	1522	1101	246	29	0.9013	0.0518	0.0375	0.0084	0.001
2	296	39298	5050	881	64	0.8620	0.1173	0.0193	0.0014	*
3	124	570	61769	5653	258	0.9034	0.0910	0.0056	*	*
4	24	154	4519	66761	802	0.9239	0.0736	0.0021	0.0003	*
5	6	34	650	7708	56688	0.8710	0.1184	0.0100	0.0005	0.0001
Test-set										
1	243	237	505	229	8	0.1989	0.1939	0.4133	0.1874	0.0065
2	79	341	1261	580	13	0.1500	0.5893	0.2551	0.0057	*
3	28	261	2854	2164	122	0.5257	0.4467	0.0279	*	*
4	18	67	1921	4348	481	0.6361	0.3514	0.0098	0.0026	*
5	4	17	383	2685	1152	0.2716	0.6331	0.0903	0.0040	0.0009

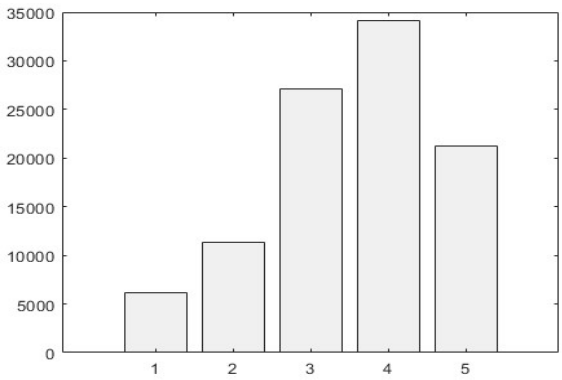
Table 3: Effect of imputation on MovieLens 1M dataset. '*' indicates that the measure is not applicable.

(a) Confusion matrix after the 1st iteration.

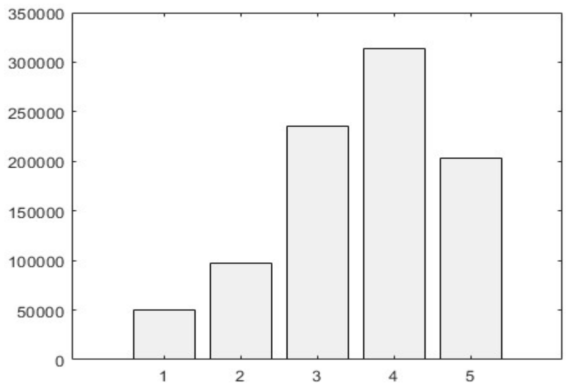
		Predicted									
		1	2	3	4	5	$HR@0$	$HR@1$	$HR@2$	$HR@3$	$HR@4$
Actual		Training-set									
	1	27187	14822	2817	111	2	0.6050	0.3298	0.0627	0.0025	0
	2	1454	47408	35883	1296	4	0.5510	0.4336	0.0151	0	$*$
	3	40	3499	166131	39166	121	0.7950	0.2042	0.0008	$*$	$*$
	4	1	41	21005	251053	7076	0.8993	0.1006	0.0001	0	$*$
	5	3	2	537	44048	136458	0.7537	0.2433	0.0030	0	0
		Test-set									
	1	3530	3530	3473	643	59	0.3142	0.3142	0.3091	0.0572	0.0053
	2	2029	5314	9988	4106	75	0.2470	0.5586	0.1909	0.0035	$*$
	3	1077	5133	23398	19980	2652	0.4479	0.4807	0.0714	$*$	$*$
	4	71	1909	17845	40173	9797	0.5756	0.3960	0.0274	0.0010	$*$
	5	32	89	4966	22351	17824	0.3938	0.4938	0.1097	0.0020	0.0007

(b) Confusion matrix after the 50th iteration.

		Predicted									
		1	2	3	4	5	$HR@0$	$HR@1$	$HR@2$	$HR@3$	$HR@4$
Actual		Training-set									
	1	84016	13840	3267	235	39	0.8286	0.1365	0.0322	0.0023	0.0004
	2	2236	100004	35825	1727	48	0.7151	0.2722	0.0123	0.0003	$*$
	3	281	4684	209436	39915	472	0.8220	0.1750	0.0030	$*$	$*$
	4	68	323	24209	287691	8165	0.8978	0.1010	0.0010	0.0002	$*$
	5	17	74	949	44973	182672	0.7988	0.1967	0.0041	0.0003	0.0001
		Test-set									
	1	3587	3502	3450	640	56	0.3193	0.3117	0.3071	0.0570	0.005
	2	1992	5418	9960	4083	59	0.2519	0.5556	0.1898	0.0027	$*$
	3	1057	5082	23525	19956	2620	0.4503	0.4793	0.0704	$*$	$*$
	4	40	1854	17799	40370	9732	0.5784	0.3945	0.0266	0.0006	$*$
	5	21	75	4886	22285	17996	0.3976	0.4923	0.1079	0.0017	0.0005



(a) MovieLens 100K



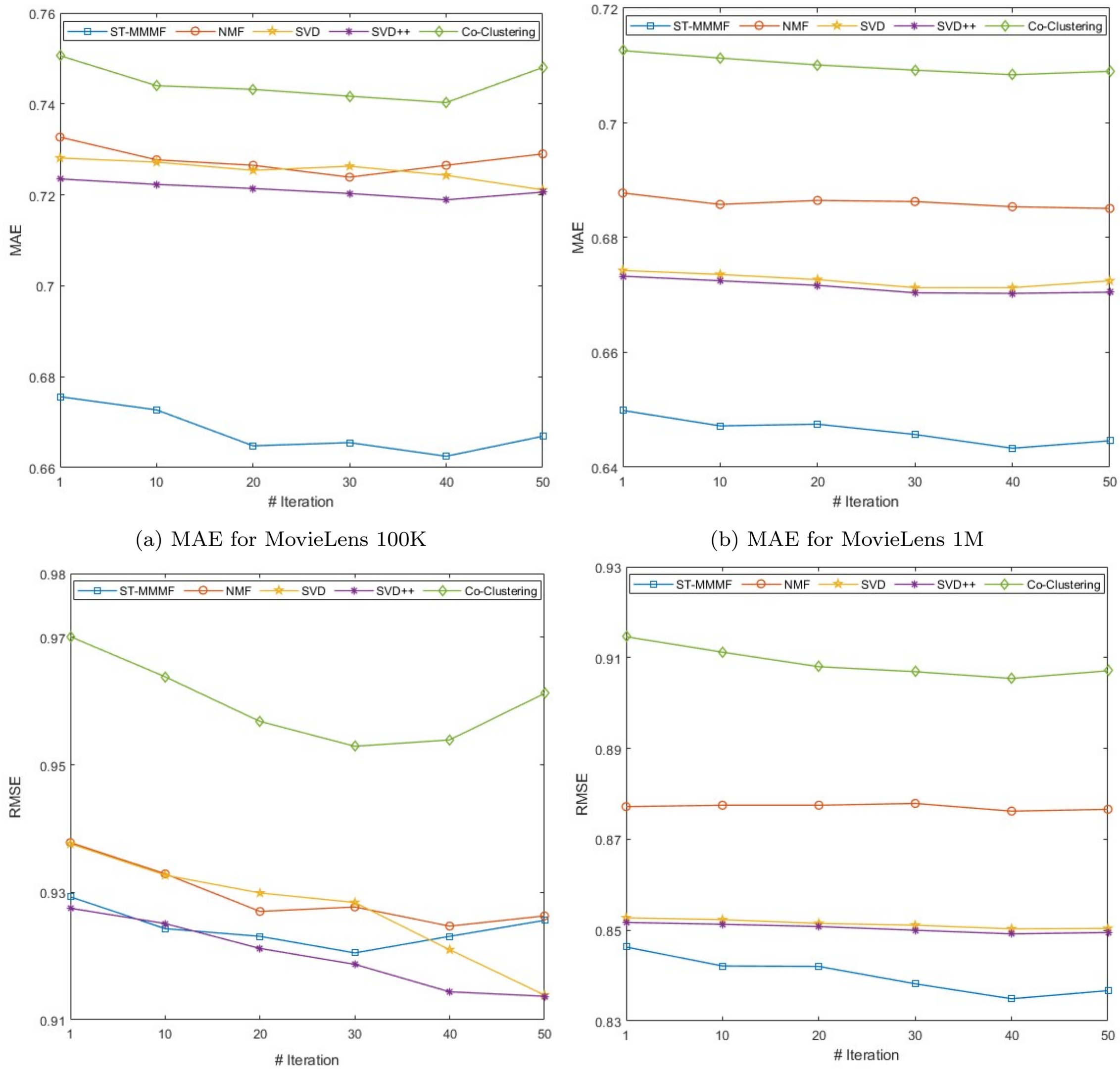
(b) MovieLens 1M

Figure 5: Rating Distribution

statistics to measure the number of times, on average, the actual and predicted rating labels differ with $\pm K$ distance. For simplicity, we denoted this as $HR@K$, where $HR@0$ for any rth rating label is the fraction of *hits* to the rth rating label in the prediction. As discussed, this statistic is more useful in discrete ordinal rating prediction. It is always better to predict '2' than '5' if the actual label is '1', unlike the multi-class classification setting where all mistakes are equal. The proposed augmentation strategy leads to a more balanced dataset and improves individual label prediction accuracy.

We next evaluated the performance of each baseline algorithm with the augmented training set. After each round of the augmentation process, the baseline models are retrained, and their performance over the test set is recorded. Figure 6 reports the results related to this experiment. It can be seen that the process of data augmentation brings substantial improvements to the performance of each baseline algorithm. Furthermore, it can also be seen that the performance of several baseline algorithms starts decreasing after a few rounds of augmentation. The reason is that test sets are created before the augmentation process begins. Hence, the proportion of the rating label in the test set nearly follows the same initial rating distribution. After several rounds of the augmentation process, the rating distribution in the training set becomes more balanced. In contrast, the rating distribution in the test set remains unchanged, leading to decreased performance over the test set. This result prompts a new line of future research of deciding the maximum tolerable percentage of ratings that can be imputed to the rating matrix.

We also experiment to validate that the proposed rating augmentation process satisfies certain desirable properties such as *monotonicity* and *invariance*. By *monotonicity*, we mean that the proposed augmentation process exhibits significant overlap among the ratings predicted with high confidence across iterations. In other words, The entries that satisfy Equation (12) in an iteration continue to be of high confidence in subsequent iterations. The augmentation process is said to be *decision invariant* if the percentage of ratings predicted with high confidence increases in each subsequent iteration. We have reported the output of the first five runs of ST-MMMF on MovieLens 100K in Table 4. It can be seen from the table that our proposed approach satisfies both the crucial properties of the augmentation



(a) MAE for MovieLens 100K

(b) MAE for MovieLens 1M

(c) RMSE for MovieLens 100K

(d) RMSE for MovieLens 1M

Figure 6: Performance of baseline algorithms on MovieLens 100K dataset

Table 4: Effect of data augmentation

# Iteration	**# Observed entries**	**# Unobserved entries**	**# Unobserved entries predicted with high-confidence**	**# Ratings augmented**	**# Overlap between the ratings predicted with high confidence**
1	80000	1506126	395851	5000	NA
2	85000	1501126	485458	5000	374637
3	90000	1496126	538751	5000	467698
4	95000	1491126	565265	5000	519462
5	100000	1486126	604835	5000	552333

process. We have observed similar performance in the refinement process.

6. Conclusions and Future Work

This paper presented a novel self-training-based semi-supervised approach, ST-MMMF, for rating augmentation and refinement that uses maximum-margin matrix factorization (MMMF) as the base learner. The proposed rating augmentation and refinement process exploits the geometrical interpretation of MMMF. The proposed approach is an iterative scheme that uses some of the ratings predicted with high confidence in every iteration to augment the training set. The ratings predicted with low confidence are removed from the training set in the refinement phase. Extensive experimental studies performed over the two highly imbalanced real-world rating datasets corroborate our claim that the proposed approach significantly reduces the prediction bias of MMMF toward rating labels of high samples. Furthermore, we also evaluated the performance of several state-of-the-art algorithms to validate that the proposed rating augmentation strategy is likely to reduce or alleviate the data sparsity problem.

There are numerous avenues for future research, addressing several issues with the proposed approach and possible modeling extension to different application areas. It is interesting to see whether ST-MMMF can be further improved by considering side information from other auxiliary domains, thereby enabling a more robust and meaningful confidence association with the predicted ratings. This line of research is more applicable to scenarios such as route recommendations that require the incorporation of multi-modal transport and locations of facilities/services (Zhang et al., 2011; Liao et al., 2013). Extension of the proposed approach to diverse applications, especially in scenarios where abundant information is unavailable for the modeling, such as course (Parameswaran et al., 2011) and tourism destination recommendations (Lucas et al., 2013), presents a compelling direction for further exploration. Furthermore, a significant issue worth exploring involves applying the proposed approach in the problem area where the observed entries do not have any predefined order. A comprehensive study in this direction is required where the objective function of the underlying maximum margin matrix factorization algorithm needs to be carefully modified to align with the principles of multi-class classification problems while preserving the maximum margin property. We plan to investigate these aspects in the future.

Acknowledgements

Vikas Kumar is supported by the Start-up Research Grant (SRG), Science and Engineering Research Board, under grant number SRG/2021/001931, UGC-BSR Start-up Grant, UGC under grant number F.30-547/2021(BSR) and the Faculty Research Programme Grant, University of Delhi, under grant number IoE/2021/12/FRP.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*, 734–749.

Bell, R., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 95–104).

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, *46*, 109–132.

Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*, .

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*, 331–370.

Chen, S., Dobriban, E., & Lee, J. H. (2020). A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, *21*, 9885–9955.

Dave, V. S., Zhang, B., Al Hasan, M., AlJadda, K., & Korayem, M. (2018). A combined representation learning approach for better job and skill recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 1997–2005).

Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., & Zhang, F. (2017). A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the AAAI Conference on artificial intelligence*. volume 31.

Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, *156*, 113748.

Gao, R., Biggs, M., Sun, W., & Han, L. (2022). Enhancing counterfactual classification performance via self-training. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 6665–6673). volume 36.

George, T., & Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 4–pp). IEEE.

Ghazanfar, M. A., & Prugel, A. (2013). The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved svd-based recommendations. *Informatica*, *37*.

Guo, G., Zhang, J., & Thalmann, D. (2014). Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems*, *57*, 57–68.

Guo, Q., Sun, Z., & Theng, Y.-L. (2019). Exploiting side information for recommendation. In *Web Engineering: 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11–14, 2019, Proceedings 19* (pp. 569–573). Springer.

He, X., Folkman, L., & Borgwardt, K. (2018). Kernelized rank learning for personalized drug recommendation. *Bioinformatics*, *34*, 2808–2816.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, .

Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, *5*, 2174.

Hwang, W.-S., Parc, J., Kim, S.-W., Lee, J., & Lee, D. (2016). "told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)* (pp. 349–360). IEEE.

Kim, D., & Yum, B.-J. (2005). Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, *28*, 823–830.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 426–434).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*, 30–37.

Krishnapuram, B., Williams, D., Xue, Y., Carin, L., Figueiredo, M., & Hartemink, A. (2004). On semi-supervised classification. *Advances in neural information processing systems*, *17*.

Kumar, V. (2019). Collaborative filtering and multi-label classification with matrix factorization. *arXiv preprint arXiv:1907.12365*, .

Kumar, V., Pujari, A. K., Sahu, S. K., Kagita, V. R., & Padmanabhan, V. (2017a). Collaborative filtering using multiple binary maximum margin matrix factorizations. *Information Sciences*, *380*, 1–11.

Kumar, V., Pujari, A. K., Sahu, S. K., Kagita, V. R., & Padmanabhan, V. (2017b). Proximal maximum margin matrix factorization for collaborative filtering. *Pattern Recognition Letters*, *86*, 62–67.

Kuo, R., Chen, C.-K., & Keng, S.-H. (2021). Application of hybrid metaheuristic with perturbation-based k-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems. *Information Sciences*, *575*, 90–115.

Liao, F., Arentze, T., & Timmermans, H. (2013). Incorporating space–time constraints and activity-travel time profiles in a multi-state supernetwork approach to individual activity-travel scheduling. *Transportation Research Part B: Methodological*, *55*, 41–58.

Liu, Z., Zheng, L., Zhang, J., Han, J., & Philip, S. Y. (2019). Jscn: Joint spectral convolutional network for cross domain recommendation. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 850–859). IEEE.

Lu, Z., Zhong, E., Zhao, L., Xiang, E. W., Pan, W., & Yang, Q. (2013). Selective transfer learning for cross domain recommendation. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (pp. 641–649). SIAM.

Lucas, J. P., Luz, N., Moreno, M. N., Anacleto, R., Figueiredo, A. A., & Martins, C. (2013). A hybrid recommendation approach for a tourism system. *Expert systems with applications*, *40*, 3532–3550.

Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, *10*, 1273–1284.

Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., & Koper, R. (2011). Recommender systems in technology enhanced learning. In *Recommender systems handbook* (pp. 387–415). Springer.

Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004, Agia Napa, Cyprus, October 25-29, 2004. Proceedings, Part I* (pp. 492–508). Springer.

Mnih, A., & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. *Advances in neural information processing systems*, *20*.

Mobasher, B., Jin, X., & Zhou, Y. (2003). Semantically enhanced collaborative filtering on the web. In *European Web Mining Forum* (pp. 57–76). Springer.

Moon, J., Jeong, Y., Chae, D.-K., Choi, J., Shim, H., & Lee, J. (2023). Comix: Collaborative filtering with mixup for implicit datasets. *Information Sciences*, *628*, 254–268.

Nartey, O. T., Yang, G., Wu, J., & Asare, S. K. (2019). Semi-supervised learning for fine-grained classification with self-training. *IEEE Access*, *8*, 2109–2121.

Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, *149*, 113248.

Ning, X., & Karypis, G. (2012). Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 155–162).

Niu, J., Wang, L., Liu, X., & Yu, S. (2016). Fuir: Fusing user and item information to deal with data sparsity by using side information in recommendation systems. *Journal of Network and Computer Applications*, *70*, 41–50.

Parameswaran, A., Venetis, P., & Garcia-Molina, H. (2011). Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, *29*, 1–33.

Ranjbar, M., Moradi, P., Azami, M., & Jalili, M. (2015). An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. *Engineering Applications of Artificial Intelligence*, *46*, 58–66.

Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., & Ré, C. (2017). Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, *30*.

Ren, Y., Li, G., Zhang, J., & Zhou, W. (2012). The efficient imputation method for neighborhood-based collaborative filtering. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 684–693).

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, .

Rennie, J. D., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning* (pp. 713–719).

Salman, K., Pujari, A. K., Kumar, V., & Veeramachaneni, S. D. (2016). Combining swarm with gradient search for maximum margin matrix factorization. In *PRICAI 2016: Trends in Artificial Intelligence: 14th Pacific Rim International Conference on Artificial Intelligence, Phuket, Thailand, August 22-26, 2016, Proceedings 14* (pp. 167–179). Springer.

Srebro, N., Rennie, J., & Jaakkola, T. (2004). Maximum-margin matrix factorization. *Advances in neural information processing systems*, *17*.

Strub, F., Gaudel, R., & Mary, J. (2016). Hybrid recommender system based on autoencoders. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 11–16).

Veeramachaneni, S. D., Pujari, A. K., Padmanabhan, V., & Kumar, V. (2019). A maximum margin matrix factorization based transfer learning approach for cross-domain recommendation. *Applied Soft Computing*, *85*, 105751.

Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. (2018). A content-based recommender system for computer science publications. *Knowledge-Based Systems*, *157*, 1–9.

Wang, Q., Yin, H., Wang, H., Nguyen, Q. V. H., Huang, Z., & Cui, L. (2019). Enhancing collaborative filtering with generative augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 548–556).

Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., & Liu, H. (2017). What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web* (pp. 391–400).

Wu, D., Shang, M., Luo, X., Xu, J., Yan, H., Deng, W., & Wang, G. (2018). Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, *275*, 180–191.

Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020a). Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, *33*, 6256–6268.

Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. (2020b). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10687–10698).

Yu, W., Lin, X., Ge, J., Ou, W., & Qin, Z. (2020). Semi-supervised collaborative filtering by text-enhanced domain adaptation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2136–2144).

Zhang, J., Liao, F., Arentze, T., & Timmermans, H. (2011). A multimodal transport network model for advanced traveler information systems. *Procedia-Social and Behavioral Sciences*, *20*, 313–322.

Zhang, Q., Lu, J., & Zhang, G. (2020). Cross-domain recommendation with multiple sources. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). IEEE.

Zhao, L., Lu, Z., Pan, S. J., Yang, Q., & Xu, W. (2016). Matrix factorization+ for movie recommendation. In *IJCAI* (pp. 3945–3951).