Improving the Estimation of Tail Ratings in Recommender System with Multi-Latent Representations

Xing Zhao, Ziwei Zhu, Yin Zhang, and James Caverlee
Department of Computer Science and Engineering, Texas A&M University
xingzhao,zhuziwei,zhan13679,caverlee@tamu.edu

ABSTRACT

The importance of the distribution of ratings on recommender systems (RS) is well-recognized. And yet, recommendation approaches based on latent factor models and recently introduced neural variants (e.g., NCF) optimize for the head of these distributions, potentially leading to large estimation errors for tail ratings. These errors in tail ratings that are far from the mean predicted rating fall out of a uni-modal assumption underlying these popular models, as we show in this paper. We propose to improve the estimation of tail ratings by extending traditional single latent representations (e.g., an item is represented by a single latent vector) with new multi-latent representations for better modeling these tail ratings. We show how to incorporate these multi-latent representations in an end-to-end neural prediction model that is designed to better reflect the underlying ratings distributions of items. Through experiments over six datasets, we find the proposed model leads to a significant improvement in RMSE versus a suite of benchmark methods. We also find that the predictions for the most polarized items are improved by more than 15%.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation system; Latent representation; Polarization; Rating distribution

ACM Reference Format:
Xing Zhao, Ziwei Zhu, Yin Zhang, and James Caverlee. 2020. Improving the Estimation of Tail Ratings in Recommender System with Multi-Latent Representations. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371810

1 INTRODUCTION

How can we predict which popular movies a user will **not** like? Or which users will like an app that has received mixed reviews? And for controversial items with polarized ratings (e.g., political books), how can we ensure that we recommend items to the right subset of users? While recommender systems (RS) have made great strides in connecting users to the right items – be it on YouTube, Yelp, Netflix, or Amazon – there are still great challenges in estimating these *tail ratings* that are far from the mean rating for many items.

We say that the *tail ratings* are ratings from a user to a specific item that are significantly lower or significantly higher than an item's average rating, typically accounting for a smaller fraction of all ratings on an item. For example, Figure 1(a) shows the rating distribution for six different data sets, all of which have a majority of ratings in the upper ranges (with a mean rating of 3.3 to 4.3). The tail ratings for Amazon Books could be defined as the ratings of 1 or 2 (significantly below the average 4.09) that account for a small fraction of all ratings. The tail ratings for MovieLens could be defined as the 1-2 ratings, as well as the 5 ratings (well below and above the average 3.35 rating). In contrast, the *head ratings* are those ratings that are close to the average rating, typically accounting for the majority of all ratings on an item.

While the importance of the distribution of ratings on RS has been long recognized, e.g., [1, 2, 15, 36], many popular methods based on latent factor models and recently introduced neural variants [3, 14, 20, 22, 25, 39] optimize for the head of these distributions, potentially leading to large estimation errors for tail ratings. As we will show in Section 3, these tail estimation errors are common across multiple domains and datasets, leading to large over-estimations of the ratings of items with very low ratings, and large under-estimations of the ratings of items with very high ratings. For example, Figure 1(c) shows large RMSE prediction errors for these tail ratings when using two popular latent factor models. These errors can lead to bad recommendations, degrade trust in the recommender, and for controversial items, potentially expose users to items they are diametrically opposed to.

In this paper, we study the problem of estimating tail ratings with an emphasis on improving the quality of these estimates within latent factor models that drive many modern recommenders. We show how many existing methods rely on an assumption of a *single latent representation* that leads to large errors in tail rating estimation. We conduct a data-driven investigation of the limitations of such an assumption underlying these models whereby ratings are assumed to fit a uni-modal distribution. Paired with this investigation, we formally analyze the limitations of these single latent representation methods with respect to tail ratings. With these limitations in mind, we propose a new method which is designed to learn *multiple latent representations* for better modeling these tail ratings. In this way, the estimation of tail ratings can escape the constraint of a uni-modal distribution. We show how to incorporate these multi-latent representations in an end-to-end neural prediction model that is designed to better reflect the underlying rating distributions of items. Through experiments over six datasets from Amazon, Goodreads, and MovieLens, we find the proposed

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371810

model leads to a significant improvement in RMSE versus a suite of benchmark methods. We also find that the predictions for the most polarized items are improved by more than 15%.

This paper is structured as follows. In Section 2, we summarize related work in latent factor models and on dealing with rating distributions in the recommendation. Section 3 introduces the datasets used in this paper, followed by a data-driven investigation and theoretical analysis of the limitations of traditional latent factor models for dealing with tail ratings. We introduce our proposed multi-latent representation approach in Section 4, and then evaluate it over multiple datasets in Section 5. Finally, we conclude our work and point out future research opportunities.

2 RELATED WORK

Ratings distributions. In terms of dealing with tail ratings, there have been a few complementary works. For example, Gediminas *et al.* investigated the impact of rating characteristics like rating density, rating frequency distribution, and value distribution, on the accuracy of popular collaborative filtering techniques [1]. Hu *et al.* observed that product ratings tend to fit a 'J-shaped' distribution [15] since users provide reviews are more likely to "brag or moan" compared to all purchasers. As an extreme case of the 'J-shaped' distribution is the 'U-shape' of *controversial items* with many extreme ratings on both sides of the distribution. Victor *et al.* in [36] formalized the concept of controversial items in recommendation systems and then compared the performance of several trust-enhanced techniques for personalized recommendations for controversial items with polarized ratings (bi-modal distribution) versus other items. Similar to our observations, they showed that predicting ratings for controversial items is much worse than for other items. Badami surveyed state-of-the-art research on the polarization [2], finding that many trust-based RS attempts to improve recommendation for controversial items by defining a trusted network for each user, e.g., [11, 27, 30, 35]. Recently, Beutel *et al.* proposed a focused learning model to improve the recommendation quality for a specified subset of items, through hyper-parameter optimization and a customized matrix factorization objective [4].

Latent factor models. Latent factor model is one of the cornerstones of RS, critical for traditional approaches [6, 19] as well as recent neural variants like NCF [14] and others [3, 14, 21, 23, 33, 39]. Furthermore, these latent factor models have been adapted in a number of directions, including location-aware recommendation systems [5, 26, 29], aspect-aware latent factor models [8], and bio-inspired approaches [31, 32], among many others. As we will demonstrate in the following section, latent factor models typically depend on an assumption of a *single latent representation.* That is, every item and user has only a single latent representation. We refer to such approaches as *Single Latent Representation* (SLR)-based methods.

	# Users	# Items	# Ratings	# Avg
Amazon Books	3,824	9,640	172,018	4.09
Amazon Digital Music	5,541	3,568	64,706	4.22
Amazon Kindle	68,223	61,934	982,619	4.35
Amazon CD & Vinyl	75,258	64,443	1,097,592	4.29
GoodReads	2,671	7,702	195,174	4.00
MovieLens	610	9,724	100,836	3.35

Table 1: All datasets have a global mean around 3.35 to 4.35, with tail ratings in the lower (1-2) and upper (5) portions of the distribution.

At the core of these latent factor models, it is assumed that both items and users live in a low-dimensional latent space, where the latent factors typically capture user preferences and item characteristics. For instance, Matrix Factorization finds the optimal low-rank matrix $P^{m \times r}$ and $Q^{n \times r}$, representing user latent factors and item latent factors respectively, such that $P \cdot Q^T$ is close to the original rating matrix $M^{m \times n}$, where r is the predefined low rank, m is the number of users, and n is the number of items.

$$arg \min_{P,Q} (M - P \cdot Q^T)^2 \tag{1}$$

If M is the user-item rating matrix, each row of P and Q corresponds to a user latent factor and an item latent factor, respectively. Since each user and item corresponds to a single row in P or Q, we say this is an SLR-based method. In practice, many latent factor models incorporate bias terms for users and items and a global offset into the prediction model. For clarity in the discussion, consider the classic matrix factorization model (MF) with bias:

$$\hat{r} = p_u \cdot q_i^T + b_u + b_i + \mu \tag{2}$$

where $\hat{r}$ is the estimated rating. In this case, p_u and q_i are the latent representations for user u and item i, respectively. The bias terms capture user bias (b_u), item bias (b_i), and a global offset (μ).

More recently, neural variants like Neural Collaborative Filtering (NCF) have been proposed to combine deep learning architectures with traditional matrix factorization [14]. In particular, NCF is structured with two sub-models: Generalized Matrix Factorization (GMF) and a Multi-Layer Perceptron. The GMF submodel corresponds to a neural version of MF, and so also relies on a single latent vector for representing a user's preference or an item's characteristics.

3 TAIL RATINGS: OBSERVATIONS AND ANALYSIS

In this section, we conduct a two-part investigation of the *single latent representation* assumption underlying latent factor models like MF and NCF and how this impacts tail ratings. Firstly (Section 3.1), we demonstrate the challenges in estimating tail ratings across six datasets that are due to the fundamental uni-modal latent factor distribution. Secondly (Section 3.2), we formally analyze the limitations of SLR-based methods with respect to tail ratings.

3.1 Data-Driven Study

We use six ratings-based datasets: Amazon Books, Amazon Digital Music, Amazon Kindle, and Amazon CDs & Vinyl [13], MovieLens [12], and GoodReads [37]. For each, we adopt the N-core selection criteria which have been shown to lead to more robust training and evaluation: that is, each user gives at least N ratings, and each item receives at least N ratings. Specifically, we use 12-core for Amazon Books and 5-core for the others. For MovieLens, we consider users who have rated at least 20 movies. For the following analyses and experiments, we randomly split the ratings of each user into training, validation, and test sets using a random 60%, 20%, 20% split. Details of these datasets are shown in Table 1.

Observations: All Ratings. For each dataset, we estimate ratings of the test set using the standard latent factor model in Equation 2

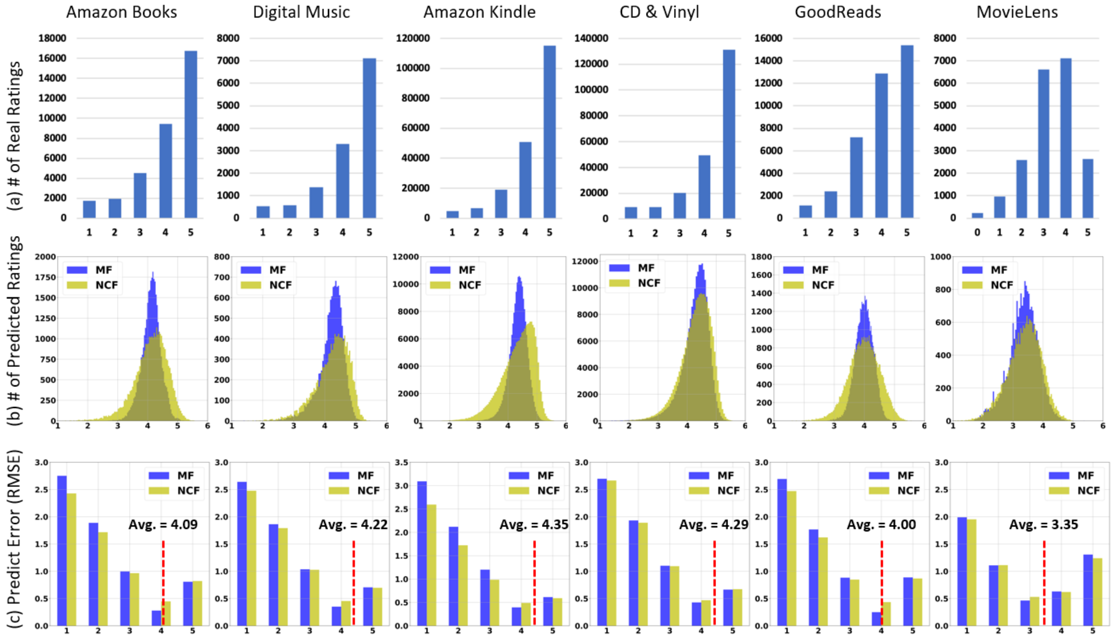

Figure 1: The count of ratings for six different datasets (a); the predicted ratings by MF and NCF are uni-modal as shown in (b); meaning that errors are concentrated in the tails of the predicted distributions (c).

(MF) and a neural variant based on NCF, since these two are foundational for both traditional and neural recommenders.

Figure 1 (a) shows the original rating distribution in the test set for each of our six datasets. As we can see, from the perspective of all ratings, the count of the original ratings fits a uni-modal distribution. The tail ratings (e.g., rating '1' and '2' for many datasets, but also rating '5' for MovieLens) only take a small portion of the overall ratings, and most ratings concentrate around the global mean (what we refer to as the *head* of the rating distribution).

Figure 1 (b) shows the distributions of predicted ratings by Matrix Factorization (blue) and Neural Collaborative Filtering (yellow). Predicted ratings by MF are normally distributed with a mean which is close to the global means in the training dataset. Regardless of the number of tail ratings in the ground truth, very few ratings are predicted around the tails. Similarly, results using the NCF are slightly better with respect to the data distribution, and the mean is slightly off-centered of the global mean. However, similar to what we observed for MF, the predicted ratings by NCF are mostly concentrated around the head of the distribution, with very few ratings predicted around the tails. These observations show how tail ratings are more likely to be under-served by traditional methods that rely on a single latent representation. Due to the high global mean, low tail ratings are most likely over-estimated by such current methods.

Figure 1 (c) demonstrates the prediction errors (RMSE) for the ground truth ratings in our test set. As we can see, the predictions are extremely poor for tail ratings for both MF and NCF. For instance, in the Amazon Books dataset, all ratings of '1', which are far from the global mean 4.09, have much worse prediction error (by both MF and NCF) than ratings '3', '4', and '5', which are closer to the global mean. Similar situations are evident for the other datasets as well. Since SLR-based models primarily under-serve these tail ratings, our goal in the following section is to improve this estimation by relaxing the single latent representation assumption.

	Controversial Items	Polarized Ratings	Rating Percentage
Amazon Books	128	1,393	0.810%
Amazon Digital Music	11	67	0.104%
Amazon Kindle	233	1,589	0.162%
Amazon CD & Vinyl	362	4,650	0.424%
GoodReads	54	752	0.385%
MovieLens	38	99	0.098%

Table 2: Ratings of controversial items in six datasets.

Observations: Polarized Ratings. We further focus on an extreme case of tail ratings: polarized ratings. Polarized ratings can indicate controversial items; a recommender that mistakenly estimates a high rating for what a user would perceive as a low rating (or vice versa) can be a serious error particularly in domains like politics [9, 16]. Following previous work [28], we adopt a variance threshold, $VAR(R_i) >= 3$, to identify items with polarized ratings. We also ensure that the items have at least a minimum number of ratings, $|R_i| >= 5$, leading to the smaller dataset in Table 2.

Focusing on two of the datasets in Figure 2, we see the original polarized distribution (green bars) of books in the test set. These distributions are bi-modal, with peaks near to the lowest rating (1) and the highest rating (5). As before, we estimate the ratings of the test set using the standard latent factor model in Equation 2 (ignoring NCF for now; the results are similar). The yellow bars in Figure 2 show the predicted ratings for these polarized books in the test set. As expected, the predicted ratings fit the uni-modal distribution and are quite distant from the original ground truth ratings. Most of the ratings on the lower end have been over-predicted into the range near to the global mean.

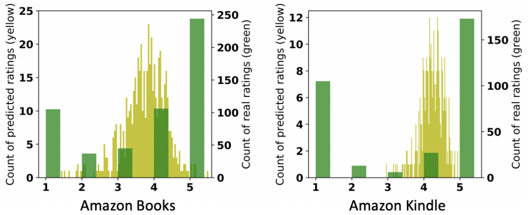

Figure 2: Rating distribution for controversial items in Amazon Books and Kindle. The actual ratings (green) for these items fit a U-shape distribution. However, the predicted ratings (yellow) for these items fit a uni-modal distribution, leading to high prediction errors.

3.2 Limitations of SLR Methods

This data-driven investigation has shown that latent factor models with a single latent representation assumption perform poorly on estimating tail ratings, and especially so for items with polarized ratings. But what is the underlying cause of these errors?

Loss Function Assumes Uni-Modal Data. From a probabilistic perspective, the prediction model found by latent factor models like the ones underlying Matrix Factorization and Neural Collaborative Filtering encourages the predicted value $F(x|\theta)$ to be close to the truth value y, where x is the given feature, and θ denotes the parameters used in the prediction function. These models typically use an L2 norm loss function, which is defined as:

$$\mathcal{L}_{SLR} = \sum_{i=1}^{N} \left\| y - F(x|\theta) \right\|_F^2 \tag{3}$$

Recall that the task of the model is to find the optimal parameter set θ using the following function:

$$\hat{\theta} = arg \min_{\theta} \mathcal{L}_{SLR} \tag{4}$$

Inside the loss function, $\left\| y - F(x|\theta) \right\|_F^2$ is the L2 norm between ground truth y and its predicted value $F(x|\theta)$. Similarly, the widely used evaluation metric, Mean Square Error (MSE), is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left\| y - F(x|\theta) \right\|_F^2 = \frac{1}{N} \mathcal{L}_{SLR} \tag{5}$$

Next, we consider a Gaussian distribution, defined as:

$$p(z|\mu, \sigma^2) = \exp\left(-\frac{\left\| \mu - z \right\|^2}{2\sigma^2} \right) \tag{6}$$

where μ is the mean and σ^2 is the variance. If we let $z = F(x|\theta)$, where $F(x|\theta)$ is the predicted value for y, let $\sigma^2 = 1$, and let $\mu = \bar{y}$ ($\bar{y}$ is the mean of data sample), then further apply the *log* to both sides, we arrive at:

$$\log p(F(x|\theta)|\mu, \sigma^2) \propto -\frac{1}{2} \left\| F(x|\theta) - \bar{y} \right\|_F^2 \tag{7}$$

which is exactly the negative of the L2 norm loss. In other words, minimizing the L2 loss (or MSE) is equivalent to maximizing the log-likelihood of a Gaussian. A similar derivation can be used to show that minimizing the L1 loss – as in mean absolute error (MAE) – is equivalent to maximizing the log-likelihood of a Laplacian.

Therefore, these traditional loss functions assume that the values y (the rating data of the original matrix M) come from a *uni-modal distribution*. So the observations in the previous section are driven by this underlying uni-modal distribution assumption. While regularization terms can be added to the loss function to help encourage the predicted values to deviate from a strict Gaussian distribution, it is still fundamentally constrained by this uni-modal distribution assumption. Indeed, any method using the L2 (or L1) norm loss – as in MF and NCF but also others – will face the same limitation. Furthermore, since the loss function forces the predicted ratings to fit a uni-modal distribution, it causes the learned latent factors to also fit a uni-modal distribution as well.

Predictions are Blurry. As we’ve seen in our previous data-driven investigation, the rating distribution for particular items is not necessarily uni-modal. As a result, SLR-based prediction models can give rise to a “blurry” problem, which occurs when the distribution of data y follows a complex distribution, e.g., a bi-modal distribution as in the polarized rating case for controversial items. In these cases, the distribution of polarized ratings p_y may consist of two Gaussian distributions, d_1 and d_2. But the distribution of the optimized predicted ratings, $p_{F(x|\theta)}$, will only fit a single uni-modal Gaussian $(d_1 + d_2)/2$. In other words, the predicted ratings tend to be blurry, which means they are forced to follow the uni-modal Gaussian, due to the average of d_1 and d_2 [7]. Hence, the predicted ratings, $p_{F(x|\theta)}$, using the single-latent-representation-based models have a uni-modal distribution, even for those items which have complex distributions, e.g., bi-modal distribution, in the training dataset.

In the context of a latent factor model like MF or NCF, we have seen that the tail ratings have worse predicted accuracy compared to ratings near to the global center (recall Figure 1). Now we have theoretically analyzed why this occurs since SLR-based models are not distribution sensitive. Regardless of how the ground truth is distributed, most predicted ratings will be in the range near to the center, and the count for each predicted range fits the uni-modal distribution. Most tail ratings will be either over-predicted or under-predicted, depending on the particular global means.

4 OUR APPROACH: MULTI-LATENT REPRESENTATION RECOMMENDER

We have experimentally and theoretically analyzed the phenomenon that ratings which are far from the center have a worse predicted error using SLR-based methods, especially for items with polarized rating distributions. In this section, we aim to overcome the limitation of *single latent representations* by proposing a new method which is aware of multi-modal rating distributions. In essence, we aim to learn *multiple latent representations* for each item and user. Our proposed method – MLR – is a neural method with two main components: a multiple latent representation factorization model (MLR-MF, refer to Section 4.1) and a gating mechanism to decide which latent representation is most appropriate (MLP-Gate, refer to Section 4.2). Together, the entire model is illustrated in Figure 3.

Setup. For a given user-item-rating dataset, our goal is to learn from the training dataset and well predict the missing ratings whose ground truth value is far from the global mean. These tail ratings are typically under-served by traditional methods. Formally speaking, in a training dataset M with potential rating range $[r_{min}, r_{max}]$ and global mean μ, we define the threshold variable $\beta = \min(|\mu - r_{min}|, |r_{max} - \mu|)$. Then we define the tail rating range, $\mathbb{T}$, is $[r_{min},$

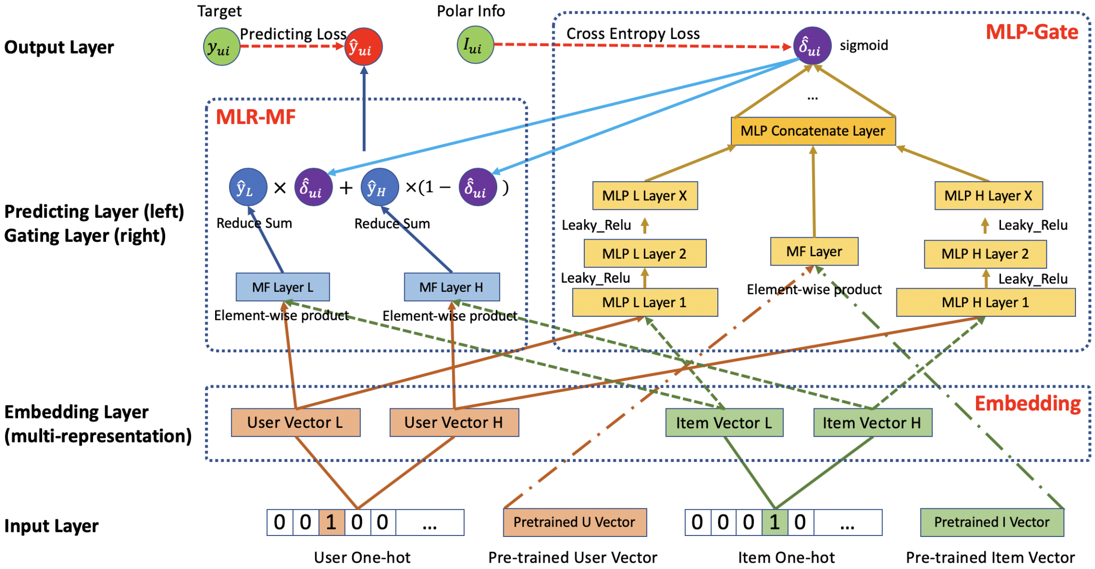

Figure 3: Overview of MLR: (1) the bottom box shows the embedding layer, where each item and user is represented as two latent vectors, L and H; (2) the top-left box shows the MLR-MF process to dot product the learned two representations for each side, L and H, respectively; and (3) the top-right box shows the MLP-Gate process which outputs the probability δ_{ui} as the gate to control which pair of representations, $p_L \cdot q_L^T$ or $p_H \cdot q_H^T$, would be used for the final prediction.

$\mu - \beta)$ if $\beta = |r_{max} - \mu|$, otherwise $\mathbb{T}$ is $(\mu + \beta, r_{max}]$. Our objective is to improve the prediction performance in the tail rating range $\mathbb{T}$, which are primarily under-served by traditional SLR-based models.

4.1 MLR Factorization Model (MLR-MF)

The fundamental principle of our approach is to model each user and item with *multiple* representations. For ease of presentation, we focus on items with bi-modal distributions, so we have two "avatars": I_{Low} and I_{High}, and each one has its latent vector as its representation. The threshold for splitting could be set as the global mean or the expectation value of all ratings for each item (or from each user) in the training dataset. Figure 4 shows the schematic of this idea. Specifically, rather than assuming the distribution of this item's ratings fits a uni-modal distribution with $\mu \approx avg$, we can instead take advantage of a mixture of Gaussian distributions $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, which would more accurately reflect the distribution of the original ratings. The absolute value of two peaks, l_1 and l_2, can systematically adjust the importance of the two representations to describe one item. That is, when an item has ratings with a uni-modal distribution (as in most cases), the splitting method would lead to $l_1 \ll l_2$ such that $N(\mu_1, \sigma_1)$ could have little influence for the majority of ratings but would still be helpful for tail ratings.

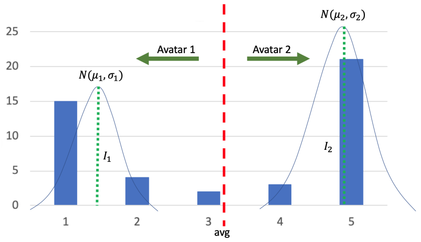

Figure 4: An example showing how to split ratings, using two "avatars" per item. In this case, two Gaussian distributions, $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, capture the bi-modal distribution for this item, where μ_1 is close to the mean of the lower ratings, whereas μ_2 is close to the mean of the higher ratings.

Of course, this bi-modal approach can be extended to consider 3, 4, or more "avatars", leading to a mixture of multiple Gaussians. In practice, however, we do find advantages to using two representations, rather than more. One of the drawbacks of the splitting process is the split matrix could be sparser than the original one; thus, there would be a sharp decline of the learning performance for each split matrix. Another reason to limit the number of mixtures is to relax the strain on the gating process (introduced in the following section) to decide which avatar is the best representation. Besides, since all our datasets have a quite small rating window (e.g., 1 to 5), few cases have visible multi-modal distribution.

In ideal conditions, the original matrix M should be automatically split to matrix M_{Low} and matrix M_{High}, where each contains only higher (or lower) ratings, with a given binary label, I, for distinguishing whether a given pair (u, i) of the original M should belong to M_{Low} or M_{High}. Next, by applying a standard SLR-based method on both new matrices, we could learn two user latent vectors P_{low} and P_{high}, and two item latent vectors Q_{Low} and Q_{High}, respectively. The ideal predicted rating, $\widetilde{r}$, for a pair (u, i) could be calculated by:

$$\widetilde{r_{ui}} = I_{ui} \times (p_{uL} \cdot q_{iL}^T) + (1 - I_{ui}) \times (p_{uH} \cdot q_{iH}^T) \tag{8}$$

meaning that, ideally, for a given user u and item i, **if we know** I_{ui}, the side (low or high) the predicted rating belongs on, then we could easily choose the $r_{ui} = p_{uL} \cdot q_{iL}^T$ or $r_{ui} = p_{uH} \cdot q_{iH}^T$. Experimental results (see Figure 5) show this ideal case can lead to an improvement in 55% of RMSE in average over the traditional latent factor model on all datasets.

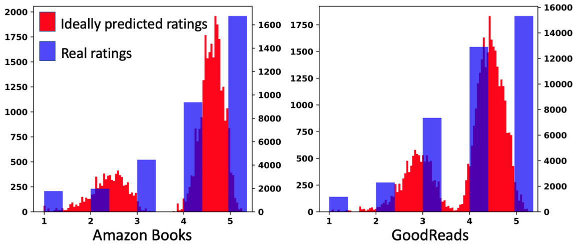

Figure 5: Ideal predicted ratings using two latent representations for both users and items, adopting the ground truth I_{ui} instead of the learned probability δ_{ui} for final prediction.

4.2 Gating Mechanism (MLP-Gate)

However, this information, I, is not known in the test dataset. Therefore, we propose to build a Gated Multi-Layer-Perceptron (MLP-Gate) to learn the gate variable δ and let it control which side (low or high) the predicted ratings should belong to. Similar to Equation 2, the final predicted rating matrix, $\hat{R}$, is calculated as follows:

$$\begin{aligned}\hat{R} =&\delta \times (P_L \cdot Q_L^T + b_{IL} + b_{UL} + \mu_L)+\\ &(1-\delta) \times (P_H \cdot Q_H^T + b_{IH} + b_{UH} + \mu_H)\end{aligned} \tag{9}$$

where b_{UL} & b_{UH} are user bias, b_{IL} & b_{IH} are item bias, and μ_L & μ_H are global means for M_{Low} and M_{High}, respectively.

The ground truth label of δ is given when we split the original matrix to two, i.e. I. Here we build a sub-model, using the learned latent representations of users (P_L and P_H) and items (I_L and I_H), to learn the δ for each given user-item pair.

From the bottom of Figure 3, in the MLR-MF learning part, each user (or item) has two latent factors, i.e. p_L and p_H (or q_L and q_H for an item). It is intuitive to combine the user feature and item feature in the same side together by concatenating *MLP_L_*1 and *MLP_H_*1. A similar design has been widely used in other neural recommenders [14, 34, 38]. For these two concatenated layers, we add the hidden layers and use a traditional Multi-layer-perceptron (MLP) to learn the interaction between user and item vectors in each side. We choose a *Leaky_Relu* as the activation function in each hidden layer. The last layer of the MLP process is named *MLP_L_X* and *MLP_H_X*, respectively.

Here, we also adopt the pre-trained latent factor, P_{MF} and Q_{MF}, by traditional SLR models. We employ the element-wise product into these two pre-trained latent factors, and concatenate *MLP_L_X* and *MLP_H_X* together, to constitute the combined layer *MLP Concatenate*. Then, we use the *sigmoid* as the activation function on *MLP Concatenate* layer and output the gate probability δ.

4.3 Learning Process

To design the loss function for the MLR, we consider two parts to monitor the performance. Firstly, we use I to check the predicted performance for δ, monitored by cross-entropy loss. Secondly, we use the ground truth rating R to check the predicted performance for the final predicted rating $\hat{R}$, by Mean Squared Error (MSE).

Specifically, the loss function $\mathcal{L}_{MLR-MF}$ for this MLR-MF sub-model is defined below:

$$\begin{aligned}\mathcal{L}_{MLR-MF} =&\|R-\hat{R}\|_F^2 + \lambda_1 \times (\|U\|_F^2 + \|I_L\|_F^2 + \|I_H\|_F^2\\ &+ \|b_u\|_F^2 + \|b_{iL}\|_F^2 + \|b_{iH}\|_F^2)\end{aligned} \tag{10}$$

where:

- R: ground truth rating matrix;
- $\hat{R}$: predicted rating matrix by Equation 9;
- λ_1: hyper-parameter controlling regularization terms for MF.

The loss function $\mathcal{L}_{MLR-MF}$ is identical to the traditional SLR-based methods. To avoid the blurry problem, we consider adding the loss for MLP-Gate together. For the part of MLP-Gate, we choose binary cross-entropy as the loss function. The loss function $\mathcal{L}_{MLR-MLP}$ is defined below:

$$\begin{aligned}\mathcal{L}_{MLR-MLP} =& -(I \times \log(\delta) + (1-I) \times \log(1-\delta))\\ &+ \lambda_2 \times (\|W\|_F^2 + \|b\|_F^2)\end{aligned} \tag{11}$$

where:

- I: ground truth label of input pair (u, i);
- δ: output value of MLP-Gate, the predicted probability of class of input pair (u, i);
- λ_2: hyper-parameter controlling the regularization terms;
- W: All weights in MLP-Gate;
- b: All bias in MLP-Gate.

Up to now, since we learn the MLR-MF and MLP-Gate simultaneously, we can combine $\mathcal{L}_{MLR-MF}$ and $\mathcal{L}_{MLR-MLP}$ as the final loss, defined as follows:

$$\mathcal{L}_{MLR} = \mathcal{L}_{MLR-MF} + \alpha \times \mathcal{L}_{MLR-MLP} \tag{12}$$

where α is used for adjusting the contribution of the loss of MLP-Gate to the total loss. As we demonstrated in Section 3.2, the L_2 norm in $\mathcal{L}_{MLR-MF}$ forces the generated ratings to be uni-modal, however, $\alpha \times \mathcal{L}_{MLR-MLP}$ could help the model to learn which pair of the latent factors, $p_L \cdot q_L^T$ or $p_H \cdot q_H^T$ could be finally adopted to calculate the predicted value, then adjust the predicted ratings to escape the uni-modal distribution.

In the learning process, we adopt the Adaptive Moment Estimation (Adam) [18] method as the optimizer to train both MLR-MF and MLP-Gate, since it yields faster convergence for both sub-models compared to SGD.

5 EXPERIMENTS

In this section, we evaluate the proposed MLR model over the six datasets listed in Table 1. We consider four scenarios: (i) an ideal case to measure the ceiling potential improvement of MLR versus SLR-based methods; (ii) a comparative study versus eight benchmark methods for all ratings (both head and tail ratings); (iii) a focused comparative study on tail ratings only; and (iv) a case study on items with extreme polarized ratings, as a special case of tail ratings. We randomly split the ratings of each user into training, validation, and test sets. 5-fold cross-validation is used in all experiments. All experimental results shown below are evaluated on the test dataset.

5.1 Ideal Results

First, we evaluate the quality of the multiple-latent-representation underlying the MLR approach in an idealized scenario. That is, we assume we have access to the ground truth label, I, from our validation dataset that determines whether a given pair (u, i) of the original M should belong to M_{Low} or M_{High}. In practice, of course, this label is unavailable to the model, but will give insights into

	NP	CoC	KNN	SlopeOne	SVD	SVD++	MF	NCF	MLR
Amazon Books	1.4779	1.0487	1.0424	1.0910	1.0787	1.0639	1.0318	1.0136	**0.9893**
Amazon Digital Music	1.4115	1.0016	0.9936	1.0585	0.9429	0.9418	0.9292	0.9369	**0.9039**
Amazon Kindle	1.2265	0.8702	0.8728	0.9106	0.8253	0.8125	0.8084	**0.7874**	0.7879
Amazon CD & Vinyl	1.3743	1.0142	1.0079	1.0659	0.9810	0.9743	0.9681	0.9597	**0.9355**
GoodReads	1.3723	0.9734	0.9732	0.9616	0.9469	0.9424	0.9386	0.9353	**0.9070**
MovieLens	1.4915	0.9864	0.9345	0.9578	0.9397	0.9355	0.9264	0.9346	**0.9155**

Table 3: Comparing MLR versus eight benchmark methods for all ratings (including both tail and head ratings). Overall, MLR shows a slight improvement in most cases, but with greater gains specifically among tail ratings (see Table 4).

the ceiling potential of MLR. This scenario corresponds to having a cross-entropy loss of 0, which is unlikely in practice.

Figure 5 shows the ideal predicted ratings using the multi-latent-factors representations in two testing datasets as examples. As we can see, the distributions are better fits for the original distribution than traditional methods (recall Figure 1), and it can capture the tail ratings far from the global centers. Overall, the RMSE is 0.5147, which is a 49.22% improvement from the best benchmark method (NCF) on the Amazon Books dataset (with an original RMSE = 1.0136). This result encourages us that MLR has a stronger and more robust representative ability than SLR. From the other perspective, the impressive ideal performance indicates that the actual performance of MLR will strongly depend on the quality of the gating mechanism in the MLP-Gate portion of the approach. Thus, the limitation now becomes how the gate δ controls which latent factors should be used to represent an item or a user.

5.2 Prediction for All Ratings

First of all, we evaluate the performance of the MLR model on all datasets comparing with other state-of-the-art methods. This evaluation considers all ratings, so improvements on tail ratings may be overshadowed by predictions on a large number of head ratings closer to the mean rating.

For comparison, we choose Normal Predictor (NP) that guesses a random rating based on the distribution of the training data, Co-clustering (CoC) [10], KNN, SlopeOne [24], SVD, SVD++ [19], Neural Collaborative Filtering (NCF), and Matrix Factorization (MF) with bias as benchmark methods. For our method, we tune the hyper-parameters in terms of number of hidden layers, number of neurons of each layer, dimensions of a representative factor, activation functions, and so on.

We first focus on the comparison of ranking quality. NDCG is the most popular measure for evaluating the ranking quality in RS [17]. Figure 6 shows the ranking quality comparison for all ratings on six datasets. We observed that the ranking performances are very close among MLR, MF, and NCF, where MLR is slightly better than MF and NCF on most datasets except Amazon Books. This result indicates that all these three methods perform well at ranking the recommended items to users. Again, our objective is to improve the explicit estimation of ratings. When the ranking qualities are similar, how good is MLR on accurately predicting user-item ratings?

Table 3 shows the overall results for all datasets using benchmark models and MLR. As we can see, MLR results in the best RMSE for five of the datasets, with a small loss to NCF on one dataset. Overall, the improvement is fairly small for all ratings, with a maximum of 3.36% improvement versus MF and 3.51% improvement for NCF. Since the non-tail ratings dominate in the aggregate, the overall improvements are small, indicating that MLR at least does not degrade rating prediction performance relative to baselines.

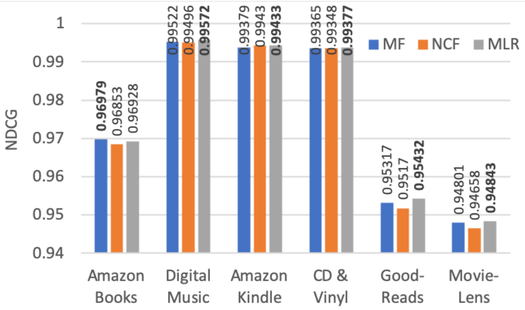

Figure 6: NDCG Comparison for all ratings on six datasets. MLR is slightly better than MF and NCF on most datasets.

5.3 Prediction for Tail Ratings

Hence, we next focus solely on the improvement to tail ratings. Table 4 shows a detailed comparison between the proposed MLR method versus NCF and MF. For each dataset, we break the predictions into buckets according to the predicted rating by MF – so there are predicted ratings from 1-2, 2-3, and so on. The first three columns show the RMSE for MF, NCF, and MLR. The columns $\Uparrow_{MLR-MF}$ and $\Uparrow_{MLR-NCF}$ show the prediction improvements for MLR versus MF and NCF for each bucket of ratings, respectively. The column Covered $\mathbb{T}$ shows the **tail ratings** that are covered by the current rating bucket. Finally, the columns $\mathbb{T} \Uparrow_{MLR-MF}$ and $\mathbb{T} \Uparrow_{MLR-NCF}$ show the improvement for MLR versus MF and NCF for each bucket of tail ratings.

We observe that for ratings far from the center, e.g., ratings less than 3, the overall improvement by MLR is substantial. For example, for Amazon Books with ratings from 1-2, MLR results in a 60%+ improvement versus both MF and NCF. For ratings from 2-3, MLR results in a 15%+ improvement versus both alternatives. Of course, these tail ratings cover a small portion of all ratings, but the improvements are large. And even for buckets with high coverage, there are still improvements (e.g., 2.54% and 6.51% for Amazon Books for rating bucket 3-4). Considering MovieLens, the tail ratings occur on both sides of the mean rating: we see large improvements for the low rating bucket 1-2 (e.g., 9% and 14% versus MF and NCF) and for the high ratings bucket 4-5 (e.g., 1.28% and 4.95% versus MF and NCF).

In some cases, there are some decreases in the predicted tail rating range near to the global mean in some datasets, e.g., a decrease of 2.59% on Amazon Books in the predicted range 4 to 5, and a decrease of 0.03% on MovieLens data in the predicted range 3 to 4, compared with NCF. In these ranges near the center, the proposed

	Predicted Range	MF (RMSE)	NCF (RMSE)	MLR (RMSE)	$\Uparrow_{MLR-MF}$ (%)	$\Uparrow_{MLR-NCF}$ (%)	Covered $\mathbb{T}$ (%)	$\mathbb{T} \Uparrow_{MLR-MF}$ (%)	$\mathbb{T} \Uparrow_{MLR-NCF}$ (%)
Amazon Books	[1,2)	0.8717	1.0625	0.3366	61.38	68.32	0.01	61.38	68.32
	[2,3)	1.5132	1.4256	1.1990	20.76	15.89	1.66	54.76	49.65
	[3,4)	1.3294	1.2752	1.2428	6.51	2.54	70.44	23.46	15.51
	[4,5)	0.8595	0.8659	0.8480	1.33	2.06	27.77	0.57	-2.59
	>5	0.1818	0.2216	0.1418	22.00	36.0	-	-	-
	All	1.0318	1.0136	0.9893	**4.11**	**2.39**	100	**15.34**	**8.74**
Movie Lens	[1,2)	1.0927	1.1561	0.9936	9.07	14.05	3.29	17.52	26.39
	[2,3)	1.0372	1.0532	1.0135	2.28	3.76	50.18	8.96	15.53
	[3,4)	0.9053	0.9083	0.8996	0.63	0.95	45.74	0.74	-0.03
	[4,5)	0.7612	0.7907	0.7515	1.28	4.95	0.79	2.73	2.54
	>5	0.8928	0.9737	0.7830	12.29	19.58	-	-	-
	All	0.9264	0.9346	0.9155	**1.17**	**2.04**	100	**3.04**	**6.16**
Amazon Kindle	[1,2)	1.6149	1.3267	1.1774	27.08	11.25	0.49	86.31	25.83
	[2,3)	1.3148	1.200	1.1962	9.01	0.34	8.81	18.36	28.14
	[3,4)	1.1055	1.068	1.0734	2.91	-0.50	59.52	13.60	11.54
	[4,5)	0.7116	0.7029	0.7024	1.29	0.06	30.70	3.57	1.69
	>5	0.4368	0.3764	0.3598	17.61	4.40	0.47	18.33	6.00
	All	0.8084	0.7874	0.7879	**2.53**	-0.01	100	**8.93**	**7.43**
CD & Vinyl	[1,2)	0.7149	1.3280	0.6660	6.84	49.85	0.03	47.70	45.29
	[2,3)	1.5414	1.6074	1.1281	26.81	29.81	2.34	48.05	50.76
	[3,4)	1.3702	1.3464	1.2804	6.54	4.90	47.14	22.12	19.78
	[4,5)	0.8755	0.8708	0.8597	1.80	1.27	50.46	5.22	2.88
	>5	0.2337	0.2592	0.2166	7.31	16.42	0.01	3.74	2.10
	All	0.9681	0.9597	0.9355	**3.36**	**2.52**	100	**11.79**	**9.50**
Good-Reads	[1,2)	-	-	-	-	-	-	-	-
	[2,3)	1.4382	1.4377	1.2134	15.62	15.59	1.90	39.62	39.35
	[3,4)	1.0570	1.0503	1.0253	3.00	2.38	82.10	10.81	8.96
	[4,5)	0.7992	0.8043	0.7729	3.90	3.30	15.99	-2.98	-4.33
	>5	0.3079	0.2911	0.3098	0.61	6.42	-	-	-
	All	0.9368	0.9353	0.9070	**3.18**	**3.02**	100	**7.88**	**6.14**
Digital Music	[1,2)	0.7059	1.7798	0.7468	-5.8	58.03	0.46	-5.8	58.03
	[2,3)	1.4220	1.5920	1.2855	9.59	19.25	10.14	31.42	44.18
	[3,4)	1.2610	1.2709	1.2134	3.77	4.52	56.68	17.19	14.88
	[4,5)	0.7974	0.7981	0.7845	1.62	1.70	32.71	1.73	1.19
	>5	0.2712	0.2915	0.2766	-1.99	5.11	-	-	-
	All	0.9292	0.9369	0.9039	**2.72**	**3.51**	100	**9.40**	**11.51**

Table 4: Comparing MLR with single latent factor models MF and NCF for different rating ranges as predicted by MF. For ratings predicted by MF and NCF, those farther from the global mean, the improvement (in the columns $\Uparrow_{MLR-MF}$ and $\Uparrow_{MLR-NCF}$) by MLR is more pronounced, especially for tail ratings (in the columns $\mathbb{T} \Uparrow_{MLR-MF}$ and $\mathbb{T} \Uparrow_{MLR-NCF}$). '-' indicates there is no predicted data in the current range by MF.

MLR approach may perform worse than traditional methods since these methods predict most ratings in this range, but MLR does not. As a result, the gating mechanism that guides δ may lead to mispredictions.

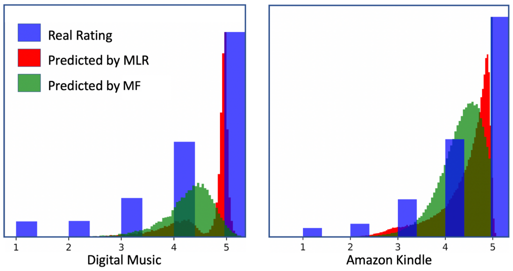

Figure 7: Distribution of real (blue) and predicted (green by MF and red by MLR) ratings for two datasets. Compared with the predicted ratings by MF (green), the predicted ratings by MLR (red) fit more accurately to the actual ratings.

For a more detailed look, Figure 7 shows the distributions of predicted ratings by MLR versus MF, as well as the ground truth ratings for Amazon Digital Music and Kindle datasets. As we can see, rather than predicting most ratings near to the global mean, the MLR approach better fits the original distribution of the ground truth data, and overcomes the uni-modal distribution limitation. Overall, we observe that our prediction model performs better on the tail ratings.

5.4 Prediction for Controversial Items

Finally, we consider a special case of tail ratings for controversial items with low polarized ratings. Specifically, we consider ratings of 1 or 2, which are over-estimated by traditional methods. Therefore, we apply the MLR model on the controversial items in three of the datasets with sufficient items. Table 5 shows the predicted performance on the low-polarized ratings (1 or 2) of the controversial items. The results are strong, where MLR improves the prediction of polarized ratings by 17.02%, 17.87%, and 15.48% on Amazon Books, Amazon Kindle, and Amazon CD & Vinyl dataset, respectively, compared with the MF method.

6 CONCLUSION

In this paper, we conduct a data-driven investigation and theoretical analysis of the challenges posed by traditional latent factor models for estimating tail ratings. These approaches assume a single latent

	Amazon Books	Amazon Kindle	Amazon CD & Vinyl
MF (RMSE)	1.97115	1.76308	1.88167
NCF (RMSE)	1.89232	1.65243	1.79345
MLR (RMSE)	1.63559	1.44794	1.59031
$\Uparrow_{MLR-MF}$	**17.02%**	**17.87%**	**15.48%**
$\Uparrow_{MLR-NCF}$	**13.57%**	**12.38%**	**11.33%**

Table 5: Comparing MLR versus baselines for controversial items.

representation, which can lead to over- and under-estimations of tail ratings, with particularly pronounced errors on controversial items. With these challenges in mind, we propose a new multi-latent representation method designed specifically to estimate these tail ratings better. Experimental results show the estimation improvement is especially strong for those items far from the ratings mean. Furthermore, the proposed model is generalizable and can be easily extended to take advantage of other SLR-based models.

In our future work, we are eager to explore the impact of incorporating additional side information into the MLR approach, particularly in the gating mechanism. This part of the model is critical to the overall success of the approach. Additional evidence like user profiles, temporal behaviors, and others could lead to even better estimation of tail ratings.

REFERENCES

[1] Gediminas Adomavicius and Jingjing Zhang. 2012. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems (TMIS)* 3, 1 (2012), 3.

[2] Mahsa Badami. 2017. Peeking into the other half of the glass: handling polarization in recommender systems. (2017).

[3] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[4] Alex Beutel, Ed H Chi, Zhiyuan Cheng, Hubert Pham, and John Anderson. 2017. Beyond globally optimal: Focused learning for improved recommendations. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 203–212.

[5] Robert P Biuk-Aghai, Simon Fong, and Yain-Whar Si. 2008. Design of a recommender system for mobile tourism multimedia selection. In *2008 2nd International Conference on Internet Multimedia Services Architecture and Applications*. IEEE, 1–6.

[6] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.

[7] Lei Cai, Zhengyang Wang, Hongyang Gao, Dinggang Shen, and Shuiwang Ji. 2018. Deep adversarial learning for multi-modality missing data completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1158–1166.

[8] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 639–648.

[9] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2016. Balancing opposing views to reduce controversy. *arXiv preprint arXiv:1611.00172* (2016).

[10] Thomas George and Srujana Merugu. 2005. A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 4–pp.

[11] Jennifer Golbeck. 2006. Generating predictive movie recommendations from trust in social networks. In *International Conference on Trust Management*. Springer, 93–104.

[12] 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.

[13] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 507–517.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[15] Nan Hu, Jie Zhang, and Paul A Pavlou. 2009. Overcoming the J-shaped distribution of product reviews. *Commun. ACM* 52, 10 (2009), 144–147.

[16] Daniel J Isenberg. 1986. Group polarization: A critical review and meta-analysis. *Journal of personality and social psychology* 50, 6 (1986), 1141.

[17] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Yehuda Koren. 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*. ACM, 426–434.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[21] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1491–1494.

[22] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.

[23] Meehee Lee, Pyungseok Choi, and Yongtae Woo. 2002. A hybrid recommender system combining collaborative filtering with neural network. In *International conference on adaptive hypermedia and adaptive web-based systems*. Springer, 531–534.

[24] Daniel Lemire and Anna Maclachlan. 2005. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 471–475.

[25] Xin Luo, MengChu Zhou, Shuai Li, Zhuhong You, Yunni Xia, and Qingsheng Zhu. 2016. A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. *IEEE transactions on neural networks and learning systems* 27, 3 (2016), 579–592.

[26] Luis Martinez, Rosa M Rodriguez, and Macarena Espinilla. 2009. Reja: a georeferenced hybrid recommender system for restaurants. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, 187–190.

[27] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 17–24.

[28] Antonis Matakos and Panayiotis Tsaparas. 2016. Temporal mechanisms of polarization in online reviews. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 529–532.

[29] Christian Matyas and Christoph Schlieder. 2009. A spatial user similarity measure for geographic recommender systems. In *International Conference on GeoSpatial Sematics*. Springer, 122–139.

[30] John O'Donovan and Barry Smyth. 2005. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*. ACM, 167–174.

[31] Lei Ren, Liang He, Junzhong Gu, Weiwei Xia, and Faqing Wu. 2008. A hybrid recommender approach based on widrow-hoff learning. In *2008 Second International Conference on Future Generation Communication and Networking*, Vol. 1. IEEE, 40–45.

[32] Tae Hyup Roh, Kyong Joo Oh, and Ingoo Han. 2003. The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert systems with applications* 25, 3, 413–423.

[33] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.

[34] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*. 2222–2230.

[35] Patricia Victor, Chris Cornelis, Martine De Cock, and Paulo Pinheiro Da Silva. 2009. Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems* 160, 10 (2009), 1367–1382.

[36] Patricia Victor, Chris Cornelis, Martine De Cock, and Ankur M Teredesai. 2009. A comparative analysis of trust-enhanced recommenders for controversial items. In *Third International AAAI Conference on Weblogs and Social Media*.

[37] Jianling Wang and James Caverlee. 2019. Recurrent Recommendation with Local Coherence. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 564–572.

[38] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 187–196.

[39] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. *arXiv preprint arXiv:1605.09477* (2016).