In2Rec: Influence-based Interpretable Recommendation

Huafeng Liu
huafeng@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Jingxuan Wen
jingxuan@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Liping Jing*
lpjing@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Jian Yu
jianyu@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Xiangliang Zhang
xiangliang.zhang@kaust.edu.sa
King Abdullah University of Science
and Technology (KAUST)
Thuwal 23955-6900, Saudi Arabia

Min Zhang
z-m@tsinghua.edu.cn
Tsinghua University
Beijing, China

ABSTRACT

Interpretability of recommender systems has caused increasing attention due to its promotion of the effectiveness and persuasiveness of recommendation decision, and thus user satisfaction. Most existing methods, such as Matrix Factorization (MF), tend to be black-box machine learning models that lack interpretability and do not provide a straightforward explanation for their outputs. In this paper, we focus on probabilistic factorization model and further assume the absence of any auxiliary information, such as item content or user review. We propose an *influence mechanism* to evaluate the importance of the users' historical data, so that the most related users and items can be selected to explain each predicted rating. The proposed method is thus called **In**fluence-based **In**terpretable **Rec**ommendation model (**In2Rec**). To further enhance the recommendation accuracy, we address the important issue of *missing not at random*, i.e., missing ratings are not independent from the observed and other unobserved ratings, because users tend to only interact what they like. **In2Rec** models the generative process for both observed and missing data, and integrates the influence mechanism in a Bayesian graphical model. A learning algorithm capitalizing on iterated condition modes is proposed to tackle the non-convex optimization problem pertaining to maximum a posteriori estimation for **In2Rec**. A series of experiments on four real-world datasets (*Movielens 10M*, *Netflix*, *Epinions*, and *Yelp*) have been conducted. By comparing with the state-of-the-art recommendation methods, the experimental results have shown that **In2Rec** can consistently benefit the recommendation system in both rating prediction and ranking estimation tasks, and friendly interpret the recommendation results with the aid of the proposed influence mechanism.

CCS CONCEPTS

• **Information systems → Collaborative filtering**; **Recommender systems**.

KEYWORDS

Recommendation system, Probabilistic matrix factorization, Collaborative Filtering, Interpretable Recommendation

ACM Reference Format:
Huafeng Liu, Jingxuan Wen, Liping Jing, Jian Yu, Xiangliang Zhang, and Min Zhang. 2019. In2Rec: Influence-based Interpretable Recommendation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3358017

1 INTRODUCTION

As an indispensable information filtering technique, recommendation system (RS) is nowadays ubiquitous in various domains, such as recommendation of products at Amazon, books/movies/music at Douban, research articles at CiteULike and so on. The system aims to provide personalized recommendation and improve user's experience. In the last few years, researchers have found that **interpretations** play a significant role in recommendation systems because it not only convinces users to accept the customized results (i.e., promotion), but also allows them to be more informed about which results to utilize (i.e., satisfaction) [24].

Significant progress has been made for recommendation. Most existing methods emphasize the optimization of performance metrics, such as Root-Mean-Square Error and Normalized Discounted Cumulative Gain, which leads to promising prediction results. However, they hardly realize the underlying reasons behind a user's decision making process, i.e., lack of interpretability. Taking the popular and powerful probabilistic matrix factorization (PMF)-based Collaborative Filtering (CF) approaches [15, 25] as examples, the missing rates can be accurately predicted with the inner product of the corresponding user factor and item factor. Unfortunately, the "latent factors" cannot be related to particular semantic concepts. It is thus difficult to understand why one item is recommended to a particular user rather than others. Recently, Abdollahi [1, 2] proposed an explainable matrix factorization (EMF) by adding an explainability regularizer to the MF-based model. Its main idea is to formulate the explainability based on the rating distribution within

[*] Liping Jing is the corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358017

the user's or item's neighborhood. This dependency on neighborhood makes EMF difficult on evaluating the explainability value for cold-start users even near-cold-start users who only have few neighborhoods and few ratings.

Moreover, most of the previous CF-based recommender systems assume that missing ratings are *missing at random* (MAR). That is, one missing rating does not depend on either the observed rating data or other missing rating data. However, users tend to only buy/watch what they are interested in, and usually rate only what they like. For many of the items that they are not interested in, users usually don't rate them and have no interaction at all, rather than giving a low rating value. It is thus inappropriate to assume that missing ratings are independent from each other and from the observed ratings. As proved in [17, 18], the recommendation data is *missing not at random* (MNAR).

We target on addressing **two important issues** in this work: the lack of interpretability of PMF-based CF methods, and the MNAR problem. For achieving the interpretability, we propose an influence mechanism in the probabilistic matrix factorization model. More specifically, besides the user and item latent factors, three more variables are introduced to capture the information for interpretation. Among them, user and item influence variables are used to indicate the importance of users' historical data. The third one is user-item relational matrix, which measures the relationship between user and item in the same latent space. With the aid of these three variables, the proposed **In**fluence-based **In**terpretable **Rec**ommendation model (**In2Rec**) has the ability to update the latent user and item factors by extracting proper amount of information from each component, and select the most related users and items to explain the predicted rating.

To handle the sparse recommendation data with missing not at random, **In2Rec** adopts a non-random missing mechanism to model the generative process for both observed and missing data. Finally, the whole recommendation procedure is formulated in a unified Bayesian graphical model, which naturally combines the influence mechanism and non-random missing mechanism. To deal with large-scale data, an efficient algorithm is designed to optimize **In2Rec** with the aid of iterated conditional modes. To the best of our knowledge, this is the first work considering the influence of historical data in interpretable recommendation. Thus, it is expected that **In2Rec** can simultaneously improve the recommendation performance and provide the interpretable results.

The rest of the paper is organized as follows. The preliminaries and related work are discussed in Section 2. Section 3 gives the proposed **In2Rec** model and the efficient scalable iterated conditional modes-based learning algorithm is given in Section 4. In Section 5, a series of experiments are presented to demonstrate the performance of **In2Rec** by comparing with the state-of-the-art recommendation methods and provide intuitive recommendation explanation for final predicted results. Finally, a brief conclusion is given in Section 6.

2 PRELIMINARIES AND RELATED WORK

Suppose there are n users and m items, $\mathbf{R} = \left[\mathbf{R}_{ij}\right]_{n\times m}$ indicates the user-item preference matrix, where the observed element $\mathbf{R}_{ij}$ records the preference data (e.g., rating score or clicks) that the i-th user gives to the j-th item. Usually, $\mathbf{R}$ is very sparse, thus, our goal is to predict the preference for unknown cell set $\Omega = \{(i,j) : \mathbf{R}_{ij} = 0\}$, i.e., $\mathbf{R}_{\Omega}$. Let $\mathbf{X} = \left[\mathbf{X}_{ij} \in \{0,1\}\right]_{n\times m}$ be an indicator matrix where $\mathbf{X}_{ij} = 1$ if $\mathbf{R}_{ij}$ is observed, and $\mathbf{X}_{ij} = 0$ if $\mathbf{R}_{ij}$ is missing.

In this section, we firstly review the existing interpretable recommendation methods in subsection 2.1. Then, the existing collaborative filtering methods working with different assumptions of missing data dependency are discussed in subsection 2.2.

2.1 Interpretable recommendation

Interpretable recommendation can provide explanation to a user what is the reason for making an item be recommended. Early approaches to personalized recommender systems mostly focused on content-based or collaborative filtering-based recommendation [3, 14, 23, 27]. Content-based recommendation attempts to model user and/or item profiles with various available content information, such as the price, color, brand of the goods in e-commerce, or the genre, director, duration of the movies in review systems [23]. Since the item contents are much understandable to the users, they are always used to explain why an item is recommended out of other candidates in content-based recommendation. However, collecting content information in different application scenarios is expensive and time-consuming. CF-based approaches, on the other hand, attempt to overcome this challenge by leveraging "wisdom of the crowds" from only rating data. The most popular CF-based recommendation includes user-based CF and item-based CF [14], which have been widely used in news and product recommendation systems. Those methods can provide intuitive interpretations, such as "*customers who bought this item also bought...*" in user-based CF and "*the item is similar to your previously loved items*" in item-based CF. Leveraging the neighborhood style explanation mechanism, Abdollahi [1, 2] presented an explainable matrix factorization (EMF) by extending MF-based CF model via an explainability regularizer. Even though EMF provides explanation, it prefers to the popular items. For the near-cold-start items with few historical data, EMF cannot perform well due to the lack of information. Inspired by the successes of attention-based deep learning due to its strong ability of representation, researchers presented neural attentive interpretable recommendation model (NAIRS) [30] to extend the traditional item similarity model [13]. In this paper, we introduce *not at random missing* mechanism to build the interpretable recommendation model so that more information can be considered.

Another stream of work seeks for recommendation interpretations from auxiliary information. For example, textual reviews and social relations are studied in additional to the basic recommender model. To make use of textual reviews, topic model is integrated with PMF to determine the explicit review-aware item features which are aligned to the latent factor for explanations generation [20, 31]. Recently, due to the powerful ability in representation learning, deep learning is adopted in recommendation to model textual content and generate the explanations [7–9, 29]. For instance, Seo et al. [29] introduced an interpretable convolutional neural network to learn the item feature from users' review information. Donkers et al. [9] combined the user-item interaction and review information in a unified LSTM framework. Moreover, social trust information has been proved an alternative view of user preference to improve trustworthiness and transparency for recommendation [22]. We in this work target on making interpretable recommendation from rating data only. The key is to infer the interpretable influence factor from observed ratings, while related work discussed here aims at linking the auxiliary information with the recommendation decisions. Our study thus has a different problem setting and is generally applicable to systems where the auxiliary information is unavailable.

2.2 Assumption of Missing Ratings

By assuming that rating data is *missing at random* (MAR), PMF-based collaborative filtering models are usually built with the only observed rating data. That is, the rating matrix $\mathbf{R}$ can be generated by an observed data model (ODM), i.e., $p(\mathbf{R}, \mathbf{Z}|\Theta_O)$ with parameter Θ_O, where $\mathbf{Z}$ is a set of latent variables. However, according to the statistical theory of missing data and the reported evidences in collaborative filtering [17, 18], the recommendation data is definitely *missing not at random* (MNAR). To consider the data with MNAR, both the observed rating matrix $\mathbf{R}$ and the missing indicator matrix $\mathbf{X}$ are generated by a missing data model (MDM), i.e., $p(\mathbf{X}|\mathbf{R}, \mathbf{Z}, \Theta_M)$ with parameter Θ_M. Consequently, the joint distribution of observed rating matrix $\mathbf{R}$, missing indicator matrix $\mathbf{X}$ and latent variables $\mathbf{Z}$ given parameters Θ_O and Θ_M can be written as $p(\mathbf{R}, \mathbf{X}, \mathbf{Z}|\Theta_O, \Theta_M) = p(\mathbf{R}, \mathbf{Z}|\Theta_O)p(\mathbf{X}|\mathbf{R}, \mathbf{Z}, \Theta_M)$.

There have been a number of methods considering MNAR data focusing on characterizing the dependency between missing data and observed rating data. Marlin et al. [19] integrated the non-random missing data mechanism into multinomial mixture model. Hernandez-lobato et al. [12] presented the first practical implementation of a probabilistic matrix factorization model for MNAR data. Bence et al. [6] and Ohsawa et al. [21] successively extended the Bayesian probabilistic matrix factorization model by introducing a priori for the missing data or considering the user's attention and item's attraction. However, these methods are characterized in a sophisticated and highly complex strategy, which will lead to time-consuming parameter estimation. In this work, we take advantage of sampling and parallel computing technique to efficiently solve the proposed model.

3 THE PROPOSED IN2REC METHOD

In this section, we present our proposed **In2Rec** model, which provides interpretable recommendation by potentially evaluating the influence of users' historical data and effectively characterizing the historical data which is missing not at random. The graphical model of **In2Rec** is given in Figure 1.

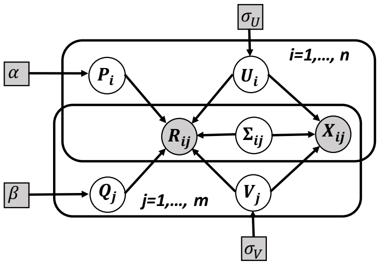


Figure 1: The graphical model of the proposed influence-based interpretable recommendation model.

3.1 Modeling the Influence of Users on Items

The existing CF-based methods usually assume that the neighbor items or users equally contribute to the current recommendation [14]. Actually, different neighbor items or users may play different roles on the prediction. As shown in [1, 2], such differences are considered to construct the explainability matrix. In **In2Rec**, thus, we introduce an influence mechanism to evaluate the importance of the users' preference data and items' historical data, so that the most relevant users and items can be selected to explain each predicted rating.

Similar to PMF, each user and item in **In2Rec** are re-represented in a common latent feature space. Meanwhile, the user latent factor ($\mathbf{U}_i \in \mathbb{R}^d$) and item latent factors ($\mathbf{V}_i \in \mathbb{R}^d$) are assumed following a *Gaussian* distribution with mean zero and co-variance matrix $\sigma_U^2\mathbf{I}$ and $\sigma_V^2\mathbf{I}$, respectively, i.e., $\mathbf{U}_i \sim \mathcal{N}(0, \sigma_U^2\mathbf{I})$ and $\mathbf{V}_j \sim \mathcal{N}(0, \sigma_V^2\mathbf{I})$. In order to make sure the predicted values $\mathbf{U}^\top\mathbf{V}$ match the observed preference data $\mathbf{R}$ as close as possible, we introduce variance matrix $\Sigma = [\sigma_{ij}]_{n\times m}$ and assume the variance variable σ_{ij} be generated from a *Laplacian* distribution with parameters γ_a and γ_b by

$$p(\sigma_{ij}) = \mathcal{L}(\sigma_{ij}|\gamma_a, \gamma_b). \tag{1}$$

To capture the influence of corresponding users and items for each rating ($\mathbf{R}_{ij}$), we introduce two influence vectors $\mathbf{P}_i \in \mathbb{R}^{m_i}$ and $\mathbf{Q}_j \in \mathbb{R}^{n_j}$ for the corresponding user i and item j, respectively. Here m_i denotes the number of preference data (number of ratings or clicks) given by user i and n_j is the number of preference data related to item j. $\mathbf{P}_{li}$ (the l-th element of $\mathbf{P}_i$) indicates the influence of l-th preference data of user i. Similarly, $\mathbf{Q}_{lj}$ indicates the influence of l-th preference data of item j. Since our goal is to build explanations by selecting the most important historical data, only the most influential preference data will be expected to have a larger influence weight, i.e., only few elements of $\mathbf{P}_i$ or $\mathbf{Q}_j$ have larger values while most of the others have smaller values. Thus, by this principle, $\mathbf{P}_{li}$ and $\mathbf{Q}_{lj}$ are assumed following *Laplacian* distribution, i.e., $\mathbf{P}_{li} \sim \mathcal{L}(\alpha_a, \alpha_b)$ and $\mathbf{Q}_{lj} \sim \mathcal{L}(\beta_a, \beta_b)$

$$\begin{aligned} p(\mathrm{P}|\alpha_a, \alpha_b) &= \prod_{i=1}^{n}\prod_{l=1}^{m_i} \mathcal{L}(\mathrm{P}_{li}|\alpha_a, \alpha_b) \\ p(\mathrm{Q}|\beta_a, \beta_b) &= \prod_{j=1}^{m}\prod_{l=1}^{n_j} \mathcal{L}(\mathrm{Q}_{lj}|\beta_a, \beta_b) \end{aligned} \tag{2}$$

where α_a and α_b are the location parameter and scale parameter of the *Laplacian* distribution for $\mathbf{P}_{li}$, respectively. Accordingly, β_a and β_b are the location parameter and scale parameter for $\mathbf{Q}_{lj}$. Given the fact that the influence vectors $\mathbf{P}_i$ and $\mathbf{Q}_j$ exist in different representation space with different dimensions, a relational matrix $\Sigma^{(ij)} \in \mathbb{R}^{m_i \times n_j}$ related to user i and item j is introduced to model the relationships between user influence and item influence, which is formulated as

$$\Sigma^{(ij)} = [\sigma_{st}]_{s\in R1(i),\, t\in R2(j)} \tag{3}$$

where $R1(i)$ (with $|R1(i)| = m_i$) is the set of items that user i rated and $R2(j)$ (with $|R2(j)| = n_j$) is the set of users who rated the item j.

Thus the confidence of observing rating $\mathbf{R}_{ij}$ can be determined with the influence vector of user i, the influence vector of item j and the corresponding relational matrix $\Sigma^{(ij)}$. The observed preference data can be generated under a *Gaussian* distribution with the aid of latent factors and influence vectors as follows,

$$p(\mathbf{R}|\mathbf{X}, \mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \Sigma) = \prod_{i=1}^{n}\prod_{j=1}^{m}[\mathbf{P}_i^\top \Sigma^{(ij)}\mathbf{Q}_j \mathcal{N}(\mathbf{R}_{ij}|\mathbf{U}_i^\top \mathbf{V}_j, \sigma_{ij})]^{\mathbf{X}_{ij}} \quad (4)$$

where $\mathbf{X} = [\mathbf{X}_{ij}]_{n\times m}$ indicates whether $\mathbf{R}_{ij}$ is observed ($\mathbf{X}_{ij} = 1$) or missing ($\mathbf{X}_{ij} = 0$). In Eq.(4), for user i who rated item j, their influence vectors $\mathbf{P}_i$ and $\mathbf{Q}_j$ are correlated through $\Sigma^{(ij)}$ controlled by the variance between the observed preference data (random variable $\mathbf{R}_{ij}$) and the predicted value (mean $\mathbf{U}_i^\top \mathbf{V}_j$). A large value of $\mathbf{P}_i^\top \Sigma^{(ij)}\mathbf{Q}_j$ indicates that rating information $\mathbf{R}_{ij}$ has a high confidence, and vice versa.

Note that only partial components of relational matrix $\Sigma^{(ij)}$ observed because there are missing values in the preference matrix $\mathbf{R}$. Thus, estimating the whole relational matrix is a challenge problem.

3.2 Modeling the Dependency of Missing Ratings

Unlike the previous work that assumes the independence of missing ratings from others, we model $\mathbf{X} = [\mathbf{X}_{ij}]_{n\times m}$ as a *Bernoulli* random variable. The derivation of $P(\mathbf{X})$ starts from modeling the discrete rating scores by a discrete *Gaussian* distribution. A principled approach to modeling discrete data is to employ a properly normalized exponential family model [4]. The probabilisitic density function of *Gaussian* distribution is an exponential family distribution and can be expressed as

$$\begin{aligned} p(x|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2}) \\ &= \exp(\frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{\mu^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})) \end{aligned} \quad (5)$$

where $\log(\sigma\sqrt{2\pi})$ is a normalizer. Denoting the preference data set by $\mathcal{Y} = \{k\}_{k=0}^{\mathcal{B}}$, where $\mathcal{B}$ is the maximum value of preference data (e.g., the maximum rating score or the maximum number of clicks in recommendation systems). To normalize the Gaussian distribution, we use $\log\sum_{k=0}^{\mathcal{B}}\exp(-\frac{1}{2\sigma^2}(k-\mu)^2)$ to replace the normalizer $\log(\sigma\sqrt{2\pi})$ in (5), then we have

$$p(x|\mu, \sigma^2) = \exp(T(x)^\top \eta(\mu, \sigma) - \log\sum_{k=0}^{\mathcal{B}}\exp(T(k)^\top \eta(\mu, \sigma))$$

where $T(x) = [(x - \frac{\mathcal{B}}{2}), (x - \frac{\mathcal{B}}{2})^2]^\top$ and $\eta(\mu, \sigma) = [\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}]^\top$.

So far, the above distribution can characterize the discrete preference data in a discrete finite domain. To introduce the influence mechanism without additional variable, we define $\mu = \mathbf{U}_i^\top \mathbf{V}_j$ and $\sigma = \Sigma_{ij}$. In this case, the discrete preference value $\mathbf{R}_{ij}$ can be generated via the following distribution

$$p(\mathbf{R}_{ij}|\mathbf{U}, \mathbf{V}, \Sigma) = \exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right) \quad (6)$$

where

$$\begin{aligned} T(\mathbf{R}_{ij}) &= \left[(\mathbf{R}_{ij} - \mathcal{B}/2), (\mathbf{R}_{ij} - \mathcal{B}/2)^2\right]^\top \\ \eta(\mathbf{U}, \mathbf{V}, \Sigma) &= \left[(\mathbf{U}_i^\top \mathbf{V}_j)/\Sigma_{ij}^2, -1/(2\Sigma_{ij}^2)\right]^\top \\ A(\mathbf{U}, \mathbf{V}, \Sigma) &= \log\sum_{k=0}^{\mathcal{B}}\exp(T(k)^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma)). \end{aligned}$$

Then, by taking a *Bernoulli* distribution, the random variable $\mathbf{X} = [\mathbf{X}_{ij}]_{n\times m}$ indicating whether $\mathbf{R}_{ij}$ is observed ($\mathbf{X}_{ij} = 1$) or missing ($\mathbf{X}_{ij} = 0$) can be modeled as

$$\begin{aligned} p(\mathbf{X}|\mathbf{R}, \mathbf{U}, \mathbf{V}, \Sigma) &= \prod_{\mathbf{X}_{ij}}\mathcal{B}\Big(\exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right)\Big) \\ &= \prod_{\mathbf{X}_{ij}=1}\Big[\exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right)\Big] \\ &\quad \times \prod_{\mathbf{X}_{ij}=0}\Big[1 - \exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right)\Big] \end{aligned}$$

3.3 The Proposed In2Rec Model

A graphical model can naturally integrate the process of modeling the influence of users on items and the dependency of missing data, as shown in Figure 1. The posterior distribution over all latent variables for **In2Rec** is given by

$$\begin{aligned} &p(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \Sigma|\mathbf{R}, \mathbf{X}, \alpha, \beta, \gamma, \sigma_U, \sigma_V) \\ &\propto p(\mathbf{R}|\mathbf{X}, \mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \Sigma)P(\mathbf{U}|\sigma_U)P(\mathbf{V}|\sigma_V) \\ &\quad \times p(\mathbf{X}|\mathbf{R}, \mathbf{U}, \mathbf{V}, \Sigma)p(\mathbf{P}|\alpha_a, \alpha_b)p(\mathbf{Q}|\beta_a, \beta_b)P(\Sigma|\gamma_a, \gamma_b) \end{aligned} \quad (7)$$

and we can get the log of the posterior distribution over latent variables as follows.

$$\begin{aligned} &\ln[p(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \Sigma|\mathbf{R}, \mathbf{X}, \alpha, \beta, \gamma, \sigma_U, \sigma_V)] \\ &= \sum_{i=1}^{n}\mathcal{N}(\mathbf{U}_i|0, \sigma_U^2\mathbf{I}) + \sum_{j=1}^{m}\mathcal{N}(\mathbf{V}_j|0, \sigma_V^2\mathbf{I}) \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{X}_{ij}[\ln \mathbf{P}_i^\top \Sigma^{(ij)}\mathbf{Q}_j \mathcal{N}(\mathbf{R}_{ij}|\mathbf{U}_i^\top \mathbf{V}_j, \Sigma_{ij}^2)] \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{m}\ln\mathcal{B}\Big(\exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right)\Big) \\ &\quad + \sum_{i=1}^{n}\sum_{l=1}^{m_i}\mathcal{L}(\mathbf{P}_{li}|\alpha) + \sum_{j=1}^{m}\sum_{l=1}^{n_j}\mathcal{L}(\mathbf{Q}_{lj}|\beta) + \sum_{i=1}^{n}\sum_{j=1}^{m}\mathcal{L}(\Sigma_{ij}|\gamma) \end{aligned} \quad (8)$$

Since the above optimization problem is difficult to solve directly, we try to find its lower bound with the aid of Jensen's inequality. Then the original optimization objective will be transferred to optimize its lower bound as follows.

$$\begin{aligned} \mathcal{J} = &-\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{2\Sigma_{ij}^2}\mathbf{X}_{ij}[\mathbf{P}_i^\top \Sigma^{(ij)}\mathbf{Q}_j(\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)^2] \\ &- \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{X}_{ij}\left[T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right] \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{m}(1 - \mathbf{X}_{ij})\ln\Big[1 - \exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)\right)\Big] \\ &- \frac{1}{2\sigma_U^2}\sum_{i=1}^{n}\mathbf{U}_i^2 - \frac{1}{2\sigma_V^2}\sum_{j=1}^{m}\mathbf{V}_j^2 - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}[\mathbf{X}_{ij}\ln\Sigma_{ij}^2] \\ &- \frac{1}{\alpha_b}\sum_{i=1}^{n}\sum_{l=1}^{m_i}|\mathbf{P}_{li} - \alpha_a| - \frac{1}{\beta_b}\sum_{j=1}^{m}\sum_{l=1}^{n_j}|\mathbf{Q}_{lj} - \beta_a| \\ &- \frac{1}{\gamma_b}\sum_{i=1}^{n}\sum_{j=1}^{m}|\Sigma_{ij} - \gamma_a| + C \end{aligned} \quad (9)$$

where C is a constant that does not depend on any parameters.

3.4 Discussion of the Interpretability of In2Rec

The main variables of **In2Rec** are $\mathbf{P}_i$, $\mathbf{Q}_j$, $\mathbf{U}_i$, $\mathbf{V}_i$ and $\Sigma^{(ij)}$. Among them, $\mathbf{U}_i$ and $\mathbf{V}_i$ are similar to the existing PMF model for latent user factor and item factor, respectively. $\mathbf{P}_i$ and $\mathbf{Q}_j$ are the new influence vectors which are not considered in the existing models.

During the process of model training, the importance of each predicted value is adaptively estimated via $\mathbf{P}_i^\top \Sigma^{(ij)} \mathbf{Q}_j$, which is different from the existing MF-based models where all rating are treated equally. Once the prediction on (ij)-th rating is made (for user i and item j), the most influential historical data (i.e., the users with larger $\mathbf{Q}_{lj}$ and items with large $\mathbf{P}_{li}$) can be selected to provide reasonable recommendation explanation. Thus, we can say the proposed method has the ability to bridge the input data and final prediction for the black-box matrix factorization model. Note that the influence of the historical data is iteratively updated along the learning process. This highlights the key difference from the existing explainable MF (EMF) [2] where the explainability value is calculated previously.

4 SCALABLE ITERATED CONDITIONAL MODES OPTIMIZATION FOR IN2REC

The optimization problem defined in Equation (9) is very likely to overfit if we cannot precisely estimate the hyperparameters, which automatically control the generalization capacity of the proposed model. Therefore, it is more desirable to estimate the parameters and hyperparameters simultaneously during model training. One possible way is to estimate each variable by its maximum a priori (MAP) value while conditioned on the rest variables and then iterate until convergence, which is also known as iterated conditional modes (ICM) [5, 10]. The ICM procedure for maximizing the objective function is presented by following steps: (1) initialize all variables and parameters; (2) Update latent variables by corresponding objective function; (3) Update hyperparameters by maximum likelihood estimation.

The values of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$, $\mathbf{Q}$ and Σ can be updated by solving the following minimization subproblems when conditioned on other variables or hyperparameters.

Update latent user factor $\mathbf{U}_i$: By fixing $\mathbf{V}$, $\mathbf{P}$, $\mathbf{Q}$ and Σ and hyperparameters, we obtain the objective function related to $\mathbf{U}_i$

$$\begin{aligned}\mathcal{J}_{U_i} = &\sum_{j=1}^{m} \frac{\mathbf{X}_{ij}}{2\Sigma_{ij}^2}[\mathbf{P}_i^\top \Sigma^{(ij)} \mathbf{Q}_j (\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)^2] \\ &+ \sum_{j=1}^{m} \mathbf{X}_{ij} \left[T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma) \right] + \frac{1}{2\sigma_U^2} \mathbf{U}_i^2 \qquad (10) \\ &- \sum_{j=1}^{m} (1 - \mathbf{X}_{ij}) \ln \left[1 - \exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma) \right) \right]\end{aligned}$$

The optimization problem (10) can be solved via stochastic gradient descent technique as follow

$$\begin{aligned}\mathbf{U}_i^{(*)} \leftarrow &\mathbf{U}_i + \gamma \Big[e_{ij} \mathbf{V}_j - \big(\mathbf{X}_{ij} + (1 - \mathbf{X}_{ij}) \frac{\exp(l_{ij})}{1 - \exp(l_{ij})} \big) \Big((\mathbf{R}_{ij} - \frac{\mathcal{B}}{2})/\Sigma_{ij}^2 \\ &- \exp(A(\mathbf{U}, \mathbf{V}, \Sigma)) \sum_{k=0}^{\mathcal{B}} ((k - \frac{\mathcal{B}}{2})/\Sigma_{ij}^2) \Big) \mathbf{V}_j - \frac{1}{\sigma_U^2} \mathbf{U}_i \Big]\end{aligned}$$

where $e_{ij} = \frac{1}{2\Sigma_{ij}^2} \mathbf{P}_i^\top \Sigma^{(ij)} \mathbf{Q}_j (\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)$ and $l_{ij} = T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma)$, and γ is the learning rate.

Update user influence vector $\mathbf{P}_i$: By fixing $\mathbf{U}$, $\mathbf{V}$, $\mathbf{Q}$ and Σ, the objective function related to $\mathbf{P}_i$ is given by

$$\mathcal{J}_{P_i} = \sum_{j=1}^{m} \frac{\mathbf{X}_{ij}}{2\Sigma_{ij}^2} [\mathbf{P}_i^\top \Sigma^{(ij)} \mathbf{Q}_j (\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)^2] + \frac{1}{\alpha_b} \sum_{l=1}^{m_i} |\mathbf{P}_{li} - \alpha_a| \qquad (11)$$

The l_1 norms $|\mathbf{P}_{li} - \alpha_a|$ in learning $\mathbf{P}_i$ can be approximated by the smoothed l_1 method [28], which is given by

$$|\mathbf{P}_{li} - \alpha_a| \approx \frac{1}{\delta} [\log(1 + \exp(-\delta(\mathbf{P}_{li} - \alpha_a))) + \log(1 + \exp(\delta(\mathbf{P}_{li} - \alpha_a)))]$$

where δ is a parameter to control approximation. Consequently, we can update $\mathbf{P}_i$ via

$$\begin{aligned}\mathbf{P}_i^{(*)} \leftarrow \mathbf{P}_i + \gamma \Big[&- \frac{1}{2\Sigma_{ij}^2} \Sigma^{(ij)} \mathbf{Q}_j e_{ij}^2 - \frac{1}{\alpha_b} \sum_{l=1}^{m_i} \big((1 + \exp(-\delta(\mathbf{P}_{li} - \alpha_a)))^{-1} \\ &- (1 + \exp(\delta(\mathbf{P}_{li} - \alpha_a)))^{-1} \big) \Big]\end{aligned}$$

We can easily speed up the optimization by updating these variables for different users in parallel. Variables related to items are item latent factors $\mathbf{V}_j$ and item influence vector $\mathbf{Q}_j$. Updating the latent item factors and the influence vectors with items can be done by an analogous strategy, which are omitted here due to page limitation.

Update variance Σ_{ij} : By fixing $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ and $\mathbf{Q}$, the objective function related to Σ_{ij} is given by

$$\begin{aligned}\mathcal{J}_{\Sigma_{ij}} = &\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{2\Sigma_{ij}^2} X_{ij} [\mathbf{P}_i^\top \Sigma^{(ij)} \mathbf{Q}_j (\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)^2] \\ &+ \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{X}_{ij} \left[T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma) \right] \\ &- \sum_{i=1}^{n} \sum_{j=1}^{m} (1 - \mathbf{X}_{ij}) \ln \left[1 - \exp\left(T(\mathbf{R}_{ij})^\top \eta(\mathbf{U}, \mathbf{V}, \Sigma) - A(\mathbf{U}, \mathbf{V}, \Sigma) \right) \right] \\ &+ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} [\mathbf{X}_{ij} \ln \Sigma_{ij}^2] + \frac{1}{\gamma_b} \sum_{i=1}^{n} \sum_{j=1}^{m} |\Sigma_{ij} - \gamma_a| \end{aligned} \qquad (12)$$

We can update Σ_{ij} using stochastic gradient descent. Meanwhile, the l_1 norm $|\Sigma_{ij} - \gamma_a|$ in learning Σ_{ij} can also be approximated by the smoothed l_1 method.

$$\begin{aligned}\Sigma_{ij}^{(*)} \leftarrow \Sigma_{ij} + \gamma \Big[&- \frac{\mathbf{P}_{ji} \mathbf{Q}_{ij}}{\Sigma_{ij}^2} (\mathbf{R}_{ij} - \mathbf{U}_i^\top \mathbf{V}_j)^2 - \mathbf{X}_{ij} \frac{1}{\Sigma_{ij}} - \frac{1}{\gamma_b} ((1 + \exp(-\delta\Sigma_{\mathrm{ij}}))^{-1} \\ &- (1 + \exp(\delta\Sigma_{\mathrm{ij}}))^{-1}) - \big(\mathbf{X}_{ij} + (1 - \mathbf{X}_{ij}) (\frac{\exp(l_{ij})}{1 - \exp(l_{ij})}) \big) \Big(- 2B(\mathbf{R}_{ij}) \\ &- \frac{1}{\exp(A(\mathbf{U}, \mathbf{V}, \Sigma))} \sum_{k=0}^{\mathcal{B}} \big(- 2 \exp(T(k)^\top A(\mathbf{U}, \mathbf{V}, \Sigma)) B(k) \big) \Big) \Big]\end{aligned}$$

where $B(x) = \Big((x - \frac{\mathcal{B}}{2}) \mathbf{U}_i^\top \mathbf{V}_j - (x - \frac{\mathcal{B}}{2})^2 \Big) / \Sigma_{ij}^3$. δ is a parameter controlling smooth approximation.

Update hyperparameters: The hyperparameters can be learned as their maximum likelihood estimates by setting their partial derivatives be zero

$$\begin{aligned} \alpha_a &= \sum_{i=1}^{n} \sum_{l=1}^{m_i} \mathbf{P}_{li} / \sum_{i=1}^{n} m_i, & \alpha_b &= \sum_{i=1}^{n} \sum_{l=1}^{m_i} |\mathbf{P}_{li} - \alpha_a| / \sum_{i=1}^{n} m_i \\ \beta_a &= \sum_{j=1}^{m} \sum_{l=1}^{n_j} \mathbf{Q}_{lj} / \sum_{j=1}^{m} n_j, & \beta_b &= \sum_{j=1}^{m} \sum_{l=1}^{n_j} |\mathbf{Q}_{lj} - \alpha_a| / \sum_{j=1}^{m} n_j \\ \gamma_a &= \sum_{i=1}^{n} \sum_{j=1}^{m} \Sigma_{ij} / mn, & \gamma_b &= \sum_{i=1}^{n} \sum_{j=1}^{m} |\Sigma_{ij} - \gamma_a| / mn \\ \sigma_U^2 &= \sum_{i=1}^{n} \mathbf{U}_i^2 / n & \sigma_V^2 &= \sum_{j=1}^{m} \mathbf{V}_j^2 / m \end{aligned} \qquad (13)$$

In the above optimizations, each variable is estimated by its maximum a priori (MAP) value while conditioned on the rest variables. These sub-optimization steps will be iterated until convergence or the maximum number of iterations reached. The whole ICM procedure is shown in Algorithm 1.

Algorithm 1 ICM Procedure for Updating Variables in **In2Rec**

```
Input: The preference data $\mathbf{R}$, initialized variables and parameters, threshold $\epsilon$
while not satisfied by the stopping condition (here, the stopping condition is that relative decrease of the objective function value is less than $\epsilon$) do
    for all users in parallel do
        Updating latent user factor $\mathbf{U}_i$ via solving (10)
        Updating user influence vector $\mathbf{P}_i$ via solving (11)
        for all items in parallel do
            Updating variance $\Sigma_{ij}$ via solving (12)
        end for
    end for
    for all items in parallel do
        Updating latent item factor $\mathbf{V}_j$ and influence vector $\mathbf{Q}_j$
    end for
    Updating hyperparameters $\{\alpha, \beta, \gamma, \sigma_U, \sigma_V\}$ via (13)
end while
Output: Updated variables ($[\mathbf{U}_i]_{i=1}^n$, $[\mathbf{P}_i]_{i=1}^n$, $[\mathbf{V}_j]_{j=1}^m$ and $[\mathbf{Q}_j]_{j=1}^m$)
```

To efficiently implement the learning process, the main part can be done in parallel. Note that in addition to using observed preference data ($\mathbf{R}_{ij} > 0$) with $\mathbf{X}_{ij} = 1$ to explain the influence mechanisms, the **In2Rec** also considers the missing data (where $\mathbf{R}_{ij} = 0$) properly to model non-random missing mechanisms. However, the number of missing data is usually almost as large as $n \times m$ due to the data sparsity, which makes it impractical for large data sets. To address this issue, we randomly sample a subset of the missing data during the optimization procedure, and alternate different subsets in different iterations. Specifically, we adopt sampling technique to select a small subset of the missing data for each user according to a fixed ratio r, which will be discussed in experimental section.

Complexity analysis: The main computation complexity of **In2Rec** is to iteratively update the variables ($\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}$ and Σ). The computational complexity to update the latent factors ($\mathbf{U}$ and $\mathbf{V}$) in one ICM iteration is $C_U = C_V = t(nnz(\mathbf{R})d + nnz(\mathbf{X})d + (nm - nnz(\mathbf{X}))dr)$, and updating influence vectors ($\mathbf{P}$ and $\mathbf{Q}$) for users and items costs $C_P = t(nnz(\mathbf{R}) + nnz(\mathbf{R})\overline{m}_i)$ and $C_Q = t(nnz(\mathbf{R}) + nnz(\mathbf{R})\overline{n}_j)$ and updating relational matrix (Σ) costs $C_\Sigma = t(nnz(\mathbf{R}) + nnz(\mathbf{X}) + (nm - nnz(\mathbf{X}))r)$ where r is the sampling ratio for unobserved preference data, t is the number of iterations for updating each variable in one ICM iteration. $\overline{m}_i$ and $\overline{n}_j$ are the average number of ratings for each user and item, respectively. Thus, the overall computational complexity is $O(T(C_U + C_P + C_V + C_Q + O_\Sigma))$, where T is the number of total iterations.

5 EXPERIMENTS

In this section, we evaluate the proposed **In2Rec** on four datasets by comparing with the state-of-the-art methods.

5.1 Experimental Setting

Datasets: Four widely used recommendation datasets, *Movielens 10M (ML 10M)* [1], *Netflix* [2], *Epinions* [3] and *Yelp*[4], are employed to test the recommendation performance. Among them, *ML 10M* and *Netflix* come from movie domain, *Epinions* belongs to online product domain, and *Yelp* is related to local business domain. The preference scores in *ML 10M* are 10 discrete numbers in the range of [0.5,5] with step 0.5, while the ratings in the other datasets are ordinal values on the scale 1 to 5, as summarized in Table 1.

Table 1: Summary of experimental datasets

	ML 10M	*Netflix*	*Epinions*	*Yelp*
♯ users (n)	71,567	480,189	49,290	1,182,626
♯ items (m)	10,681	17,770	139,738	156,638
♯ ratings	10,000,054	100,000,000	664,824	4,731,265
RDensity	1.31%	1.17%	0.010%	0.0026%
$\overline{m}_i$	143	208	14	4
$\overline{n}_j$	936	5,627	5	30

$\overline{m}_i$: the average number of items rated by each user
$\overline{n}_j$: the average number of users interested in each item

Metrics for Interpretability: Two metrics, Explainability Precision (EP) and Explainability Recall (ER), are adopted to evaluate model interpretability [1]. EP is defined as the proportion of explainable items in the top-n recommendation list relative to the number of recommended (top-n) items for each user. Similar to the recall metric, ER is the proportion of explainable items in the top-n recommendation list relative to the number of all explainable items for a given user. Note that an item j is explainable for user i if $\mathbb{E}(\mathbf{R}_{ij} | j \in N_p) = \sum_{k=0}^{\mathcal{B}} k \times \frac{N_p \cap I_{i,k}}{|N_p|} > \theta$, where N_p is the set of similar items to item p and $I_{i,k}$ is the set of items that were given rating k by user i. Following [1], we set explainable threshold $\theta = 0.01$.

Metrics for Prediction: Two well-known evaluation metrics, (MAE $= \sum_{(i,j) \in R_t} |\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}| / |R_t|$) and root mean square error (RMSE $= \sqrt{\sum_{(i,j) \in R_t} (\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij})^2 / |R_t|}$), are adopted to evaluate the rating prediction accuracy, where R_t is the testing set. $\mathbf{R}_{ij}$ and $\hat{\mathbf{R}}_{ij}$ are the ground truth and predicted rating given by the i-th user to the j-th item. Smaller RMSE and MAE values indicate better result. For ranking estimation, we use Recall ($Re@K(i) = |Re(i) \cap T(i)| / |T(i)|$), where $Re(i)$ denotes the set of recommended items to user i and $T(i)$ denotes the set of favorite items of user i. Meanwhile, the normalized discounted cumulative gain (NDCG) is adopted to measure the item ranking accuracy of different algorithms, which can be computed by: $NDCG@K(i) = DCG@K(i)/IDCG@K(i)$ (where $DCG@K(i) = \sum_{j \in \Omega(i)} (2^{\mathbf{R}_{ij}} - 1)/\log_2(j+1)$, and $IDCG$ is the DCG value with perfect ranking). $\Omega(i)$ denotes the set of items that user i has rated in testing set. Five-fold cross-validation technique is used and their averaged results are reported.

Baselines: We compare the recommendation accuracy of **In2Rec** with nine existing methods. Among them, PMF [26] and BPMF [26] are low-rank matrix approximation model. MNAR-PMF [12], MNAR-BPMF [12] and GPMF [21] are CF algorithm considering non-random missing mechanisms. EMF is an explainable matrix factorization model with explainability regularizer. NAIRS [30] is a

[1] https://grouplens.org/datasets/movielens/
[2] https://www.netflixprize.com
[3] http://www.trustlet.org/downloaded_epinions.html
[4] https://www.yelp.com/dataset/challenge

neural attentive interpretable recommendation method. NCF [11] and Mult-VAE [16] are neural network-based CF methods.

Parameter setting: The parameters of all the compared algorithms are adopted either from their original papers or determined by experiments. For smooth l_1 strategy, we adopt $\delta = 5000$ to control smooth approximation. In iterated conditional model-based learning, we adopt $\epsilon = 0.00001$ as the convergence threshold and $T = 300$ as the maximum number of iterations. We investigate the recommendation performance of **In2Rec** with ranks in {5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 } and find the best value is $d = 20$ for *ML 10M*, $d = 40$ for *Netflix*, $d = 10$ for *Epinions* and *Yelp*. We adopt sampling technique to select a small subset of unobserved missing data for each user according to a fixed ratio r (from 0.1 to 1 stepping by 0.1) and find the performance is better when the sampling ratio is in [0.3,0.5], where the running time is acceptable.

5.2 Results and Discussion

In this subsection, we investigate the proposed model **In2Rec** from three facets. Firstly, **In2Rec** is compared with state-of-the-art collaborative filtering based recommendation methods in terms of interpretability, rating prediction and ranking estimation. Secondlly, we show that the proposed **In2Rec** has the ability to understand model behavior. Finally, the explainable recommendation results are shown with the aid of influence mechanisms.

5.2.1 Performance Evaluation. The first experiment is designed to evaluate **In2Rec** by comparing with several baselines. Three kind of tasks, interpretability, rating prediction and ranking estimation, are adopted to evaluate the recommendation performance and interpretability of the proposed method and the existing methods.

Mean EP (MEP) and Mean ER (MER) are reported in Table 2, which are the average values of EP and ER over all users. Note that among the compared algorithms, PMF, BPMF and EMF are standard CF algorithm, MNAR-PMF, MNAR-BPMF and GPMF are CF algorithms considering non-random missing mechanisms, and NCF, Mult-VAE and NAIRS are deep learning based CF recommendation methods. The best and second results are marked in bold and underlined, respectively. As expected, **In2Rec** outperforms other methods in terms of MEP and MER, which indicates that the proposed model has the ability to improve top-n recommendations in terms of the explainability of the suggested list. Among all the baselines, EMF achieves competitive performance since introducing explainability regularizer into MF-based CF model has the ability to consider neighborhood style explanations.

We select the most competitive interpretable method EMF for comparison, the interpretability performance on each rating frequency group of four datasets is shown in Table 3 (in terms of MEP@10). The proposed **In2Rec** and EMF have the similar trends with respect to different item rating frequency, i.e., MEP@10 becomes better and better with the increasing of item rating frequency, which indicates that item rating frequency plays an important role in recommendation performance. To be exciting, the proposed **In2Rec** outputs the highest quality for all rating frequency groups on four datasets. Obviously, **In2Rec** has significant improvements compared with EMF, especially on items with low item rating frequency. This result further demonstrates that **In2Rec** has the ability to effectively handle data with different item rating frequencies, especially for near-cold-start items.

Table 4 shows the recommendation performance between **In2Rec** and several baselines on rating prediction task. As expected, **In2Rec** significantly improves the recommendation performance. In the baselines, the methods with non-random missing data (MNAR-PMF, MNAR-BPMF, GPMF) outperform the methods with random missing data assumption (EMF), which indicates non-random missing assumption is helpful to capture missing mechanisms and improve the recommendation accuracy. Moreover, deep learning based CF algorithms (NCF and NAIRS) also achieve competitive performance due to the advantage in feature learning. **In2Rec** is superior to all baselines, which demonstrates that discrete preference distribution has the ability to well model the missing mechanisms and improve recommendation performance.

Besides the evaluation on testing rating prediction, we also investigate the performance of ranking prediction. Table 5 lists the Recall and NDCG of **In2Rec** by comparing nine baselines on ranking estimation for two representative datasets (*ML (10M)* and *Yelp*). Note that we only kept users, who rated at least 5 items, and items, which were rated by at least 5 users. This operation keeps 58,057 users and 7,223 items (density= 0.978%) for *ML (10M)*), and 204,512 users and 152,370 items (density =0.101%) for *Yelp*. As shown in the results, **In2Rec** can also achieve higher item ranking accuracy than the other compared algorithms due to the proper non-random missing assumption. Those experiments demonstrate that **In2Rec** can not only provide accurate rating prediction but also correctly achieve item ranking for each user. Similar results are obtained for other two datasets.

5.2.2 Understanding Model Behavior. By telling us the training points "*responsible*" for a given prediction, influence vectors reveal insights about how models rely on and extrapolate from the training data. In this section, we show that **In2Rec** can understand model behavior with the aid of influence mechanism.

To better understand how users/items influences work in the proposed model **In2Rec**, we investigate the properties of the learned user influence matrix **P** and item influence matrix **Q**. Specifically, we split rating data into ten groups at interval of 0.1 according to their influence value $\mathbf{P}_{li}$ or $\mathbf{Q}_{lj}$. Figure 2 shows the proportion of user and item influence value in difference intervals. Obviously, they approximately follow *Gamma-like* distribution. We can see that most of user influence values belong to $[0.2, 0.5]$. which indicates a large number of ratings have modest influence for final prediction, while a few ratings have large influence value ($\mathbf{P}_{li} \in (0.9, 1]$). In these four datasets, the ratings with large user influence in *ML 10M*, *Netflix*, *Epinions* and *Yelp* are 168067, 41869, 6468, 1020408 respectively, and they are about 1.681%, 1.02%, 0.973%, and 0.885% of all ratings in the corresponding datasets, which indicates only a few part of ratings dominate the final prediction. Moreover, we demonstrate the rating distribution for users with large user influence value ($\mathbf{P}_{li} \in [0.9, 1]$). Figure 3 shows the rating distribution for users with large user influence obtained via **In2Rec** and the original rating distribution. As we can see, rating distributions have the similar trends with respect to different rating values in the same dataset, which indicates that influential ratings obtained by **In2Rec** have the ability to reflect the whole rating properties.

Table 2: Comparing different methods in terms of explainability metrics (MEP and MER).

Datasets	*ML 10M*		*Netflix*		*Epinions*		*Yelp*	
Metrics	MEP@10	MER@10	MEP@10	MER@10	MEP@10	MER@10	MEP@10	MER@10
PMF	0.5834	0.0049	0.6322	0.0089	0.3451	0.0025	0.3342	0.0325
BPMF	0.6316	0.0051	0.6531	0.0091	0.3416	0.0025	0.3457	0.0351
MNAR-PMF	0.6353	0.0052	0.6618	0.0092	0.3651	0.0027	0.3759	0.0353
MNAR-BPMF	0.6355	0.0052	0.6621	0.0092	0.3658	0.0027	0.3788	0.0354
GPMF	0.6361	0.0053	0.6589	0.0091	0.3564	0.0026	0.3742	0.0349
EMF	0.6431	0.0053	0.6657	0.0093	0.3699	0.0028	0.3923	0.0357
NCF	0.5862	0.0049	0.6414	0.0090	0.3544	0.0026	0.3761	0.0352
Mult-VAE	0.5811	0.0048	0.6325	0.0089	0.3597	0.0026	0.3778	0.0353
NAIRS	0.6344	0.0051	0.6455	0.0090	0.3589	0.0036	0.3833	0.0355
In2Rec	**0.6854**	**0.0054**	**0.6711**	**0.0094**	**0.3711**	**0.0028**	**0.4126**	**0.0359**

Table 3: Comparisons of EMF and the proposed In2Rec on all items with different rating degrees in terms of MEP@10.

Datasets	Methods	0-5	6-10	11-20	21-50	51-100	101-200	> 200
	EMF	0.5623	0.5811	0.6004	0.6124	0.6211	0.6331	0.6428
ML 10M	**In2Rec**	0.6432	0.6568	0.6633	0.6694	0.6733	0.6754	0.6852;
	Improve	14.36%	13.01%	10.47%	8.40%	5.22%	6.68%	6.59%
	EMF	0.5812	0.5915	0.6195	0.6352	0.6531	0.6657	0.6654
Netflix	**In2Rec**	0.6182	0.6234	0.6424	0.6533	0.6654	0.6705	0.6710
	Improve	6.36%	5.39%	2.29%	2.85%	1.88%	0.72%	0.84%
	EMF	0.3697	0.3695	0.3793	0.3851	0.3825	0.3914	0.3998
Epinions	**In2Rec**	0.3710	0.3708	0.3805	0.3861	0.3834	0.3920	0.3903
	Improve	0.35%	0.35%	0.32%	0.27%	0.26%	0.17%	0.14%
	EMF	0.3921	0.3924	0.3968	0.3989	0.4026	0.4105	0.4129
Yelp	**In2Rec**	0.4125	0.4126	0.4133	0.4024	0.4173	0.4234	0.4241
	Improve	5.20%	5.14%	4.26%	3.65%	4.16%	3.78%	3.29%

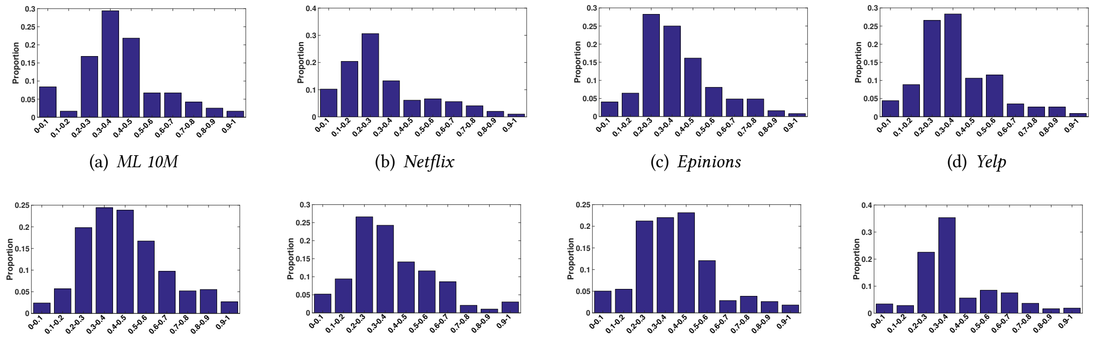
Table 4: Comparing different recommendation methods for rating prediction task.

Datasets	*ML 10M*		*Netflix*		*Epinions*		*Yelp*	
Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MNAR-PMF	0.6123	0.7912	0.6141	0.8193	1.0743	1.3153	0.6631	0.8402
MNAR-BPMF	0.6042	0.7933	0.6085	0.8121	1.0746	1.3169	0.6595	0.8376
GPMF	0.6089	0.7983	0.6115	0.8162	1.0672	1.3116	0.6615	0.8389
EMF	0.6431	0.8247	0.6385	0.8516	1.1124	1.3563	0.6684	0.8557
NCF	0.6033	0.7910	0.6124	0.8178	1.0592	1.3022	0.6655	0.8424
NAIRS	0.6122	0.7915	0.6173	0.8215	1.0754	1.3173	0.6616	0.8394
In2Rec	**0.5921**	**0.7842**	**0.5997**	**0.8064**	**1.0421**	**1.2913**	**0.6315**	**0.7965**

Table 5: Comparing different recommendation methods for ranking estimation task on *ML 10M* and *Yelp* dataset.

Dataset	*ML 10M*				*Yelp*			
Metrics	Re@5	Re@10	NDCG@5	NDCG@10	Re@5	Re@10	NDCG@5	NDCG@10
MNAR-PMF	0.5341	0.5433	0.6953	0.7538	0.4796	0.5185	0.6923	0.7306
MNAR-BPMF	0.5431	0.5463	0.6988	0.7574	0.4836	0.5218	0.6929	0.7315
GPMF	0.5496	0.5589	0.7012	0.7593	0.4837	0.5214	0.6937	0.7324
EMF	0.4932	0.5031	0.6937	0.7493	0.4626	0.4873	0.6902	0.7285
NCF	0.6521	0.6634	0.7064	0.7601	0.4901	0.5278	0.6932	0.7314
NAIRS	0.5437	0.5469	0.6984	0.7468	0.4833	0.5213	0.6911	0.7302
In2Rec	**0.6983**	**0.7071**	**0.7131**	**0.7659**	**0.5061**	**0.5398**	**0.6953**	**0.7348**

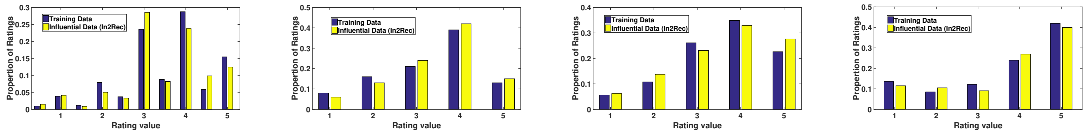
For further investigate the reasonability of learned influence value, we evaluate the recommendation performance by modifying training data with different influence levels. Specifically, for each user, all ratings related to this user are ranked according to their user influence value in an descending order, and then split into several subsets at interval of 10. In each experiment, we remove different subset and record the recommendation accuracy. Table 6 lists the Recall and NDCG when removing different rating set with different influence value, where G_k indicates the rating set from which k-th subset for each user is removed. Intuitively, the users' ratings with large influence should have large impact toward recommendation, and the ratings with small influence have less impact toward final prediction. As shown in Table 6, **In2Rec**-G_1 has a greatest degree of loss in ranking accuracy since we remove the top-10 influential ratings for each user in G_1. The lower the influence value of the removed data is, the higher the prediction accuracy is. Similarly, the item influence value $\mathbf{Q}_{Ij}$ has the same properties, we omit it due to the page limitation. Therefore, based on these observations, we



(a) *ML 10M* (b) *Netflix* (c) *Epinions* (d) *Yelp*

(e) *ML 10M* (f) *Netflix* (g) *Epinions* (h) *Yelp*

Figure 2: User influence value distribution ((a) to (d)) and Item influence value distribution ((e) to (h)) obtained via In2Rec



(a) *ML 10M* (b) *Netflix* (c) *Epinions* (d) *Yelp*

Figure 3: Rating distribution for users with large user influence value on (a) *ML 10M*, (b) *Netflix*, (c) *Epinions* and (d) *Yelp*.

can say that **In2Rec** has the ability to trace the influential rating data and explain model behaviors.

Table 6: The effect of influential data points for users on *ML 10M* and *Yelp* (e.g., In2Rec-G_1 indicates the results obtained via In2Rec on training set by removing influential set G_1).

Datates	*ML 10M*		*Yelp*	
Metrics	Re@10	NDCG@10	Re@10	NDCG@10
In2Rec	**0.7071**	**0.7659**	**0.5398**	**0.7348**
In2Rec-G_1	0.6422	0.6538	0.4943	0.7198
Setback	9.18%	14.64%	8.43%	2.04%
In2Rec-G_2	0.6738	0.6841	0.5135	0.7253
Setback	4.71%	10.68%	4.87%	1.29%
In2Rec-G_3	0.6791	0.6901	0.5233	0.7284
Setback	3.96%	9.90%	3.06%	0.87%
In2Rec-G_4	0.6854	0.6954	0.5294	0.7305
Setback	3.07%	9.20%	1.93%	0.59%

5.2.3 Interpretable results. Finally, we demonstrate the utility of our interpretable recommendation. The proposed **In2Rec** has the ability to tell users what previous behavior data affect his or her decision according to user influence vector $\mathbf{P}_i$ and which neighbors' behavior data influence users' final prediction based on item influence vector $\mathbf{Q}_j$. For example, If system gives a prediction $\hat{\mathbf{R}}_{25}$ to user 2 on item 5, the influential behavior data can be illustrated to the user to support the final prediction, as shown in Figure 4. Here we define user i likes item j when $\mathbf{R}_{ij} \geq 4$, otherwise user i dislikes item j. Similarly, $\mathbf{P}_{li} \geq 0.8$ or $\mathbf{Q}_{lj} \geq 0.8$ indicates strong influence.

Consequently, in order to investigate how the learned influence values are relevant to the inherent properties of users and items, we analyze the influence value from four different perspectives. Particularly, five representative groups are selected from users and items including (*influential user, impressionable user*, *independent users*, *popular item*, and *long-tail item*), as shown in Table 7.

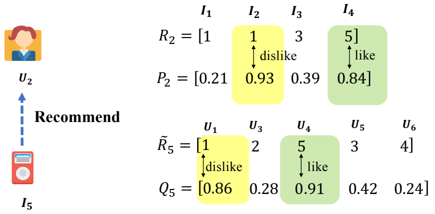

I_1 I_2 I_3 I_4
R_2 = [1 1 3 5]
dislike
like
P_2 = [0.21 0.93 0.39 0.84]
U_2
Recommend
U_1 U_3 U_4 U_5 U_6
$\tilde{R}_5$ = [1 2 5 3 4]
dislike
like
Q_5 = [0.86 0.28 0.91 0.42 0.24]
I_5

Figure 4: *An example of recommendation result: From* P_2*, we can see* I_4 *and* I_2 *play important roles on the preference of user* U_2 *(like and dislike respectively). Similarly, according to* Q_5*,* U_4 *and* U_1 *are the dominant users for item* I_5 *(like and dislike respectively).*

Table 7: Five different kinds of role defined in In2Rec.

Roles	Definition
influential user	user who has strong influence to other users
impressionable user	user who is influenced by other users
independent user	user who takes his/her own decision
popular item	item which attracts more users
long-tail item	item which attracts less users

The Pearson's moment coefficient of Skewness $\mathcal{S}$ in probability theory and statistics is used to discover the properties of the distribution of $\mathbf{P}$ or $\mathbf{Q}$,

$$\mathcal{S}(Y) = \mathbb{E}\Big[\big(\frac{Y-\mu}{\sigma}\big)^3\Big] = \frac{\mu_3}{\sigma^3} \tag{14}$$

where μ_t is the t-th central moment and σ is the standard deviation. Note that we set mean μ to 0.5 since all of the influence values are located in range [0, 1].

Specifically, given a user i, we can capture the Skewness of the distribution of $\{\mathbf{Q}_{i\cdot}\}$ =$\{\mathbf{Q}_{ij} | j \in R(i)\}$, where $R(i)$ indicates the set of items that the user i rated. All users are ranked according to their rating frequency in an descending order. If $\mathcal{S}(\mathbf{Q}_{i\cdot}) < 0.5$ and the number of items that user i rated belongs to the top 20%, user i will be labeled with *influential user*. Similarly, all items are ranked according to their rating frequency in an descending order. Given an item j, if $\mathcal{S}(\mathbf{P}_{j\cdot}) < 0.5$ and the number of users who rated item j belongs to the top 20%, item j will be a *popular item*. Considering whether the user i is *impressionable user* or not, we can get the influence value set $\{\,\mathbf{Q}_{k\cdot}\,\}$, where k is the index of users who rated the same item with user i. If $\mathcal{S}(\mathbf{Q}_{k\cdot}) < 0.5$, user i will be assigned with a label *impressionable user*. *Independent user* can be recognized via the value of $\mathbf{P}_{\cdot i}$ and $\mathbf{Q}_{i\cdot}$. If $\mathcal{S}(\mathbf{P}_{\cdot i}) < 0.5 \wedge \mathcal{S}(\mathbf{Q}_{i\cdot}) > 0.5$, user i will be an *independent user*. For long-tail items, they are ranked according to their rating frequency in an ascending order. Items that belong to the top 20% and $\mathcal{S}(\mathbf{P}_{j\cdot}) > 0.5$ are selected as the *long-tail items*.

Based on the definition of different groups, we can obtain the group information, as shown in Table 8. Due to the page limitation, we only list the results on *ML 10M* and *Yelp*.

Table 8: Summarization of five user/item groups obtained via In2Rec from *ML 10M* and *Yelp*.

Datasets	*ML 10M*	*Yelp*
♯ *influential user*	467 (0.65%)	15,847 (1.34%)
♯ *impressionable user*	3,030 (4.22%)	74,032 (6.26%)
♯ *independent user*	2,383 (3.33%)	17,148 (1.45%)
♯ *popular item*	179 (1.68%)	3,994 (2.55%)
♯ *long-tail item*	2,419 (22.65%)	52,144 (33.29%)

Based on the assigned labels, we can get more intuitive recommendation explanation as below

> To U_2:
>
> We recommend I_5 to you because you like *long-tail item* I_4 *and dislike popular item* I_2*, Meanwhile, the influential user* U_4 *likes popular item* I_5 *and independent user* U_1 *dislikes* I_5.

6 CONCLUSION

In this paper, we propose an Influence-based Interpretable Recommendation model (**In2Rec**), which has the ability to evaluate the importance of the both users' and items' historical data by introducing influence mechanism and non-random missing mechanism, so that the most related users and items can be selected to explain each predicted result. An ICM-based learning algorithm is proposed to handle the non-convex optimization problem pertaining to **In2Rec**. The experimental results on four real-world datasets have shown that **In2Rec** can effectively improve the interpretability and recommendation accuracy in both rating prediction and ranking estimation tasks and friendly provide interpretable results.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61822601, 61773050, 61672311 and 61632004; the Beijing Natural Science Foundation under Grant Z180006; the Beijing Municipal Science & Technology Commission under Grant Z181100008918012.

REFERENCES

[1] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable matrix factorization for collaborative filtering. In *Proceedings of WWW*. 5–6.

[2] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using explainability for constrained matrix factorization. In *Proceedings of ACM RecSys*. ACM, 79–83.

[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on KDE* 6 (2005), 734–749.

[4] Erling Bernhard Andersen. 1970. Sufficiency and exponential families for discrete sample spaces. *J. Amer. Statist. Assoc.* 65, 331 (1970), 1248–1255.

[5] Julian Besag. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)* (1986), 259–302.

[6] Bence Bolgar and Peter Antal. 2016. Bayesian matrix factorization with non-random missing data using informative gaussian process priors and soft evidences. In *Proceedings of ICPGM*. 25–36.

[7] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of WSDM*. ACM, 108–116.

[8] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Visually explainable recommendation. *arXiv preprint arXiv:1801.10288* (2018).

[9] Tim Donkers, Benedikt Loepp, and JÃijrgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of ACM RecSys*. 152–160.

[10] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)* (1989), 271–279.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of WWW*. 173–182.

[12] J. M. HernÃąndez-Lobato, N. Houlsby, and Z. Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *Proceedings of ICML*. 1512–1520.

[13] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of ACM SIGKDD*. ACM, 659–667.

[14] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to usenet news. *Cacm* 40, 3 (1997), 77–87.

[15] Y. Koren, R. Bell, and C. Kolinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[16] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of WWW*.

[17] Roderick JA Little and Donald B Rubin. 2014. *Statistical analysis with missing data*. Vol. 333. John Wiley & Sons.

[18] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[19] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of ACM RecSys*. 5–12.

[20] Julian Mcauley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of ACM RecSys*. 165–172.

[21] Shohei Ohsawa, Yachiko Obara, and Takayuki Osogami. 2016. Gated probabilistic matrix factorization: learning users' attention from missing values. In *Proceedings of IJCAI*. AAAI Press, 1888–1894.

[22] Haekyu Park, Hyunsik Jeon, Junghwan Kim, Beunguk Ahn, and U Kang. 2018. UniWalk: explainable and accurate recommendation for rating and network data. In *Proceedings of SIAM CDM*. SIAM.

[23] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The Adaptive Web*. Springer, 325–341.

[24] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender systems handbook* (2nd ed.).

[25] R. Salakhutdinov and A. Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of NIPS*. 1257–1264.

[26] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of ICML*. ACM, 880–887.

[27] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW*. ACM, 285–295.

[28] Mark Schmidt, Glenn Fung, and Rmer Rosales. 2007. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Proceeding of ECML*. Springer, 286–297.

[29] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of ACM RecSys*. 297–305.

[30] Shuai Yu, Yongbo Wang, Min Yang, Baocheng Li, Qiang Qu, and Jialie Shen. 2019. NAIRS: A neural attentive interpretable recommendation system. In *Proceedings of the ACM WSDM*. ACM, 790–793.

[31] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of ACM SIGIR*. ACM, 83–92.