

Discrete Trust-aware Matrix Factorization for Fast Recommendation

Guibing Guo¹, Enneng Yang^{1*}, Li Shen^{2*}, Xiaochun Yang¹ and Xiaodong He³

¹Northeastern University, China

²Tencent AI Lab, China

³JD AI Research, China

guogb@swc.neu.edu.cn, 1871149@stu.neu.edu.cn, mathshenli@gmail.com,
yangxc@mail.neu.edu.cn, xiaodong.he@jd.com

Abstract

Trust-aware recommender systems have received much attention recently for their abilities to capture the influence among connected users. However, they suffer from the efficiency issue due to large amount of data and time-consuming real-valued operations. Although existing discrete collaborative filtering may alleviate this issue to some extent, it is unable to accommodate social influence. In this paper we propose a discrete trust-aware matrix factorization (DTMF) model to take dual advantages of both social relations and discrete technique for fast recommendation. Specifically, we map the latent representation of users and items into a joint hamming space by recovering the rating and trust interactions between users and items. We adopt a sophisticated discrete coordinate descent (DCD) approach to optimize our proposed model. In addition, experiments on two real-world datasets demonstrate the superiority of our approach against other state-of-the-art approaches in terms of ranking accuracy and efficiency.

1 Introduction

Trust-aware recommender systems have received much attention recently due to their abilities of capturing the influence among connected users. Trust data is a good supplement to the sparse rating data and furthermore provides more accurate prediction rating [Ma *et al.*, 2008; Mohsen and Martin, 2010; Yang *et al.*, 2013; Guo *et al.*, 2015]. In addition, collaborative filtering (CF) have been widely used in trust-aware recommender systems due to their simplicity. However, such an approach often suffers from serious efficiency dilemma on large data sets. Specifically, the main task in a recommendation system is to recommend top-k items to meet users' favorite interests, in which it requires a huge amount of computations to calculate the user's rating of each item and then rank it according to the rating.

Recently, the discrete matrix factorization technique captures attention due to its recommendation efficiency. It adopts hash technology [Zhang *et al.*, 2016; Zhang *et al.*, 2017;

Lian *et al.*, 2017; Liu *et al.*, 2018] to map latent features of users and items to a joint hamming space and transforms the items recommendation task into a similarity search problem, which greatly improves the recommendation efficiency. Specifically, binary code can be executed with bit operation. At the same time, binary code can be searched quickly: Since the users and the items are in the joint hamming space, when recommending items for a user, it is only necessary to find the items with a small hamming distance around the user, this makes our recommendation time independent of the number of items. In addition, thanks to the binary code, we need less memory to store trained latent feature of users and items.

By incorporating trust-aware and hash technology, we propose two new models, called DTMF model and DTMF-D model, to inherit the dual advantages of CF-based trust-aware recommendation and discrete matrix factorization, *i.e.*, high accuracy and fast efficiency. To the best of our knowledge, this is the first work to perform hash technique on matrix factorization for trust-aware recommendation¹. Specifically, the challenges and our contributions are briefly summarized as:

- The first challenge is how to effectively learn binary code from real-values based trust-aware matrix without sacrificing too much information. We direct learning the latent feature of truster and trustees, and items as binary codes via discrete optimization. By adding balanced and decorrelated constraints for each bits, we are able to represent richer information in more compact code.
- The second challenge is how to efficiently optimize the proposed new discrete matrix factorization models which consider extra trust relationships among each user. We adopt an efficient discrete optimization algorithm to solve each subproblems alternately. This alternating optimization technique could constantly reduce the quantization loss from real value to discrete value.
- We demonstrate the superiority of proposed models and discrete optimization methods on two real-world datasets against other state-of-art models and solvers. Preliminary experiments indicate that the jointly proposed models and discrete optimization methods can simultaneously improve the recommendation efficiency and ensure the predicted accuracy.

¹We acknowledge a concurrent work [Liu *et al.*, 2019], which has not appeared online when we submitted our work.

*Corresponding author

2 Related Work

The performance and efficiency of recommendation system are two important indexes in practical application, there's been a lot of research on this, can be summarized as the following trust-based Recommendation and discrete hash:

2.1 Trust-based Recommendation

Social information based recommendation system has been widely studied, the basic idea of the model is to promote the user to employ sharing rating data and trust data simultaneously. [Ma *et al.*, 2008] firstly proposed probability matrix factorization to solve the data sparsity and poor prediction accuracy problem by adopting rating data and user's social network data. [Mohsen and Martin, 2010] believed that the rating of a trusted user should directly affect the latent feature of the user, rather than the final rating of user. Therefore, they proposed that the latent feature of users should be close to the average of all the latent feature of users he trusts. [Ma *et al.*, 2011] systematically illustrated how to design the objective function of a social regularization. [Yang *et al.*, 2013] proposed a hybrid model that combines both a truster model and a trustee model, in which a user should consider both the influence of his truster users and the influence of his trustee users. [Guo *et al.*, 2015] taken into account both the explicit and implicit influences of rating and trust information in predicting ratings. As mentioned above, this type of recommendation suffered from the efficiency issue on large datasets.

2.2 Discrete Hashing

Hashing technique is an efficient approach for fast similarity search and its detailed description can be founded in [Wang *et al.*, 2016]. The pioneer work [Abhinandan *et al.*, 2007] generated hash code based on the user's history behavior by using Locality-Sensitive Hashing algorithm. [Zhou and Zha, 2012] proposed to learn binary code to tackle the efficiency problem of making recommendations in the case of large users and items. [Zhang *et al.*, 2014] adopted hashing technique and considered that recommendation should be focused on users' preference over items rather than their similarities, so they propose a novel model to solve the challenge. To summarize these before work, learning hash code can be regarded as two independent stages: relaxed learning and binary quantization. Due to the large quantization accuracy loss in the two stages discrete optimization algorithm, [Zhang *et al.*, 2016] first transformed the two-stage discrete optimization into a joint optimization and by introducing the balanced and decorrelated constraints, more abundant information can be learned with less bit. In addition, some work has added some auxiliary information to the discrete collaborative filtering. [Zhang *et al.*, 2017] used hashing technique to personalized recommendations from implicit feedback, [Lian *et al.*, 2017] used textual content information to discrete matrix factorization whose model extended the recommendation task from regression to classification.

To the best of our knowledge, all these discrete collaborative filtering algorithms ignored the important social relationships. However, trust information could contain a strong correlation with taste, it has been independently exploited to

improve the predictive performance. Our model fills in the gap of the discrete model and trust-aware model by simultaneously learning binary code of truster and trustee.

3 Our Model

For better readability, we first denote some necessary notations. Generally, we use bold uppercase and lowercase letters to represent matrices and vectors, respectively; and adopt non-bold letters for scalars. In particular, \mathbf{A} represents a matrix with a suitable dimension, where \mathbf{a}_i is the i -th column vector and A_{ij} is the (i, j) -th component of matrix \mathbf{A} . Besides, we use $\|\mathbf{A}\|_F$ and $tr(\mathbf{A})$ to denote the Frobenius norm and trace of matrix \mathbf{A} , respectively.

3.1 DTMF Model

In our DTMF model, we denote $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m] \in \{\pm 1\}^{r \times m}$ and $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n] \in \{\pm 1\}^{r \times n}$ as r -length ($r \ll m, n$) binary representation of users and items respectively, where m, n is the number of users and items. Thus a user's preference for an item can be measured by the common bits [Zhou and Zha, 2012] of their latent feature vector in Hamming space. We define the Hamming similarity between user i and item j as follows:

$$\begin{aligned} \text{sim}_H(\mathbf{b}_i, \mathbf{d}_j) &= 1 - \frac{1}{r} \text{dist}_H(\mathbf{b}_i, \mathbf{d}_j) = \frac{1}{r} \sum_{k=1}^r I(b_{ik} = d_{jk}) \\ &= \frac{1}{2r} \left(\sum_{k=1}^r I(b_{ik} = d_{jk}) + r - \sum_{k=1}^r I(b_{ik} \neq d_{jk}) \right) \\ &= \frac{1}{2r} \left(r + \sum_{k=1}^r b_{ik} d_{jk} \right) = \frac{1}{2} + \frac{1}{2r} \mathbf{b}_i^\top \mathbf{d}_j, \quad (1) \end{aligned}$$

where $I(b_{ik}, d_{jk})$ is 1 when $b_{ik} = d_{jk}$ is true, otherwise is 0. $\text{dist}_H(\mathbf{b}_i, \mathbf{d}_j)$ is the hamming distance between \mathbf{b}_i and \mathbf{d}_j . In other words, for an observed rating $S_{ij} \in [0, 1]$, we can transform it to $S_{ij} \leftarrow 2rS_{ij} - r$ such that the inner product $\mathbf{b}_i^\top \mathbf{d}_j$ can be used to approximate such a value by Eq. (1).

For trust information, let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m] \in \{\pm 1\}^{r \times m}$ be the binary vectors of trustees, i.e., those trusted by some other users. Likewise, the inner product $\mathbf{b}_i^\top \mathbf{w}_t$ of user i and trustee t can be used to approximate an observed trust value $T_{it} \in [0, 1]$ after being transformed by $T_{it} \leftarrow 2rT_{it} - r$ according to Eq.(1).

Finally, the main goal of our work is to reconstruct the interactions of both user-item ratings and user-user trust as accurately as possible, whereby the binary (and compact) representation of users and items can be learned for an efficient recommendation.

In addition, we follow the suggestions of [Zhou and Zha, 2012] to further consider the balanced and de-correlated constraints, in order to maximize the information binary codes can represent. Specifically, the balanced constraint applies the principle of maximum entropy of uniformly distributed information in information theory, while the de-correlated constraint contends that each bit should be as independent as possible. To sum up, we define our objective function as

shown in Eq.(2), given by:

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{B}, \mathbf{D}} &= \sum_{i,j \in \Omega} (S_{ij} - \mathbf{b}_i^\top \mathbf{d}_j)^2 + \lambda \sum_{i,k \in \Gamma} (T_{it} - \mathbf{b}_i^\top \mathbf{w}_t)^2 \\
 \text{s.t. } & \mathbf{B} \in \{\pm 1\}^{r \times m}; \mathbf{D} \in \{\pm 1\}^{r \times n}; \mathbf{W} \in \{\pm 1\}^{r \times m}; \\
 & \underbrace{\mathbf{B}\mathbf{1}_m = 0; \mathbf{D}\mathbf{1}_n = 0; \mathbf{W}\mathbf{1}_m = 0;}_{\text{Balanced}} \\
 & \underbrace{\mathbf{B}\mathbf{B}^\top = m\mathbf{I}_r; \mathbf{D}\mathbf{D}^\top = n\mathbf{I}_r; \mathbf{W}\mathbf{W}^\top = m\mathbf{I}_r,}_{\text{Decorrelated}}
 \end{aligned} \quad (2)$$

where λ is a trade-off parameter to indicate the importance of trust reconstruction for overall rating prediction.

Note that it is NP-hard to directly solving problem (2). For the ease of problem solving, we soften the balanced and de-correlated constraints by defining three sets $\mathcal{B}, \mathcal{D}, \mathcal{W}$ so as to minimize their distances to $\mathbf{B}, \mathbf{D}, \mathbf{W}$. Take \mathcal{B} as an example since other variables basically follow the same definitions. Specifically, it is defined as $\mathcal{B} = \{\mathbf{X} \in R^{r \times m} | \mathbf{X}\mathbf{1}_m = 0; \mathbf{X}\mathbf{X}^\top = m\mathbf{I}_r\}$, and the distance as $\text{dist}(\mathbf{B}, \mathcal{B}) = \min_{\mathbf{X} \in \mathcal{B}} \|\mathbf{B} - \mathbf{X}\|_F$. Therefore, the objective function Eq.(2) can be rewritten as follows:

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{B}, \mathbf{D}, \mathbf{W}} & \sum_{i,j \in \Omega} (S_{ij} - \mathbf{b}_i^\top \mathbf{d}_j)^2 + \lambda \sum_{i,t \in \Gamma} (T_{it} - \mathbf{b}_i^\top \mathbf{w}_t)^2 \\
 & + \alpha \text{dist}^2(\mathbf{B}, \mathcal{B}) + \beta \text{dist}^2(\mathbf{D}, \mathcal{D}) + \gamma \text{dist}^2(\mathbf{W}, \mathcal{W}) \\
 \text{s.t. } & \mathbf{B} \in \{\pm 1\}^{r \times m}; \mathbf{D} \in \{\pm 1\}^{r \times n}; \mathbf{W} \in \{\pm 1\}^{r \times m}; \\
 & \mathbf{X}\mathbf{1}_m = 0; \mathbf{Y}\mathbf{1}_n = 0; \mathbf{Z}\mathbf{1}_m = 0; \\
 & \mathbf{X}\mathbf{X}^\top = m\mathbf{I}_r; \mathbf{Y}\mathbf{Y}^\top = n\mathbf{I}_r; \mathbf{Z}\mathbf{Z}^\top = m\mathbf{I}_r,
 \end{aligned}$$

where $\alpha, \beta, \gamma > 0$ are used to indicate the degree of proximity between the two matrices. As $\mathbf{B}^\top \mathbf{B}$ and $\mathbf{X}^\top \mathbf{X}$ are constant, we can use $2\alpha \text{tr}(\mathbf{B}^\top \mathbf{X})$ to substitute $\|\mathbf{B} - \mathbf{X}\|_F$ for optimization, which also holds for \mathbf{D} and \mathbf{W} . We derive the following optimization function:

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{B}, \mathbf{D}, \mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}} & \sum_{i,j \in \Omega} (S_{ij} - \mathbf{b}_i^\top \mathbf{d}_j)^2 + \lambda \sum_{i,t \in \Gamma} (T_{it} - \mathbf{b}_i^\top \mathbf{w}_t)^2 \\
 & - 2\alpha \text{tr}(\mathbf{B}^\top \mathbf{X}) - 2\beta \text{tr}(\mathbf{D}^\top \mathbf{Y}) - 2\gamma \text{tr}(\mathbf{W}^\top \mathbf{Z}) \\
 \text{s.t. } & \mathbf{B} \in \{\pm 1\}^{r \times m}; \mathbf{D} \in \{\pm 1\}^{r \times n}; \mathbf{W} \in \{\pm 1\}^{r \times m}; \\
 & \mathbf{X}\mathbf{1}_m = 0; \mathbf{Y}\mathbf{1}_n = 0; \mathbf{Z}\mathbf{1}_m = 0; \\
 & \mathbf{X}\mathbf{X}^\top = m\mathbf{I}_r; \mathbf{Y}\mathbf{Y}^\top = n\mathbf{I}_r; \mathbf{Z}\mathbf{Z}^\top = m\mathbf{I}_r.
 \end{aligned}$$

3.2 DTMF-D Model

At the same time, we also propose an alternative way to use trust data termed as the DTMF-D model. Its basic idea is to impose that two users with trust relationship should be in a closed position in hamming space, that is, to minimize the hamming distance between two trust users $\text{dist}_H(b_i, b_t)$.

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{B}, \mathbf{D}} & \sum_{i,j \in \Omega} (S_{ij} - \mathbf{b}_i^\top \mathbf{d}_j)^2 + \lambda \sum_{i,t \in \Gamma} \text{dist}_H(\mathbf{b}_i, \mathbf{b}_t) \\
 \text{s.t. } & \mathbf{B} \in \{\pm 1\}^{r \times m}; \mathbf{D} \in \{\pm 1\}^{r \times n}; \\
 & \mathbf{B}\mathbf{1}_m = 0; \mathbf{D}\mathbf{1}_n = 0; \mathbf{B}\mathbf{B}^\top = m\mathbf{I}_r; \mathbf{D}\mathbf{D}^\top = n\mathbf{I}_r.
 \end{aligned}$$

We soften the balanced and de-correlated constraints and adopt Eq. (1) to compute the hamming distance. In this way,

we can formulate the following objective function:

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{B}, \mathbf{D}, \mathbf{X}, \mathbf{Y}} & \sum_{i,j \in \Omega} (S_{ij} - \mathbf{b}_i^\top \mathbf{d}_j)^2 + \lambda \sum_{i,t \in \Gamma} \left(\frac{r}{2} - \frac{1}{2} \mathbf{b}_i^\top \mathbf{b}_t \right) \\
 & - 2\alpha \text{tr}(\mathbf{B}^\top \mathbf{X}) - 2\beta \text{tr}(\mathbf{D}^\top \mathbf{Y}) \\
 \text{s.t. } & \mathbf{B} \in \{\pm 1\}^{r \times m}; \mathbf{D} \in \{\pm 1\}^{r \times n}; \\
 & \mathbf{X}\mathbf{1}_m = 0; \mathbf{Y}\mathbf{1}_n = 0; \mathbf{X}\mathbf{X}^\top = m\mathbf{I}_r; \mathbf{Y}\mathbf{Y}^\top = n\mathbf{I}_r.
 \end{aligned}$$

We present DTMF-D to show how trust can be utilized in other manners for discrete collaborative filtering, but till far we have found that DTMF generally gets better performance than the others through extensive experimental study. We do not show the detailed results in Section 4, but instead focus more on the evaluation of our main approach DTMF.

3.3 Optimization

We use an alternative optimization technique to solve the DTMF models, which alternately update the variables $\mathbf{B}, \mathbf{D}, \mathbf{W}, \mathbf{X}, \mathbf{Y}$ and \mathbf{Z} . The detailed algorithm description is presented in Algorithm 1. Below, we show that each subproblem has a closed-form solution.

B-subproblem

In this subproblem, we aim to optimize \mathbf{B} with fixed $\mathbf{D}, \mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$. We can parallelly update each column \mathbf{b}_i of \mathbf{B} according to the following binary optimization:

$$\begin{aligned}
 \operatorname{argmin}_{\mathbf{b}_i \in \{\pm 1\}^r} & \mathbf{b}_i^\top \left(\sum_{j \in \Omega_i} \mathbf{d}_j \mathbf{d}_j^\top \right) \mathbf{b}_i - 2 \left(\sum_{j \in \Omega_i} S_{ij} \mathbf{d}_j^\top \right) \mathbf{b}_i - 2\alpha \mathbf{x}_i^\top \mathbf{b}_i \\
 & + \lambda \mathbf{b}_i^\top \left(\sum_{t \in \Gamma_i} \mathbf{w}_t \mathbf{w}_t^\top \right) \mathbf{b}_i - 2\lambda \left(\sum_{t \in \Gamma_i} T_{it} \mathbf{w}_t^\top \right) \mathbf{b}_i,
 \end{aligned}$$

where Ω_i, Γ_i are the set of rated items and trusted users of user i , respectively.

Due to the existence of discrete constraints, such a problem becomes NP-hard to solve. Instead, we use Discrete Coordinate Descent (DCD) to update \mathbf{b}_i bit to bit. Let b_{ik} be the k -th coordinate in \mathbf{b}_i and $b_{i\bar{k}}$ be the rest of \mathbf{b}_i excluding b_{ik} . We then update \mathbf{b}_i bit by bit using the following formula:

$$\begin{aligned}
 b_{ik} & \leftarrow \text{sgn}(K(\check{b}_{ik}, b_{ik})) \\
 \check{b}_{ik} & = \sum_{j \in \Omega_i} (S_{ij} - \mathbf{d}_{j\bar{k}}^\top \mathbf{b}_i^\top \bar{\mathbf{b}}_i) \mathbf{d}_{jk} + \alpha x_{ik} \\
 & + \lambda \sum_{t \in \Gamma_i} (T_{it} - \mathbf{w}_t^\top \bar{\mathbf{b}}_i^\top \bar{\mathbf{b}}_i) \mathbf{w}_{tk},
 \end{aligned} \quad (3)$$

where $\text{sgn}(\cdot)$ is a sign function, $K(\check{b}_{ik}, b_{ik})$ is a function such that $K(\check{b}_{ik}, b_{ik}) = b_{ik}$ if $b_{ik} \neq 0$, and $K(\check{b}_{ik}, b_{ik}) = \check{b}_{ik}$ otherwise. It does not update b_{ik} when b_{ik} is equal to 0.

D-subproblem

In this subproblem, we optimize \mathbf{D} with fixed $\mathbf{B}, \mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$. Similar to the B-subproblem, we parallelly update \mathbf{D} according to

$$\operatorname{argmin}_{\mathbf{d}_j \in \{\pm 1\}^r} \mathbf{d}_j^\top \left(\sum_{i \in \Omega_j} \mathbf{b}_i \mathbf{b}_i^\top \right) \mathbf{d}_j - 2 \left(\sum_{i \in \Omega_j} S_{ij} \mathbf{b}_i^\top \right) \mathbf{d}_j - 2\beta \mathbf{y}_j^\top \mathbf{d}_j.$$

We also use Discrete Coordinate Descent (DCD) to update

d_j bit to bit, and each bit d_{j_k} based on the following rule:

$$\begin{aligned} d_{j_k} &\leftarrow \text{sgn}(K(\check{d}_{j_k}, d_{j_k})) \\ \check{d}_{j_k} &= \sum_{i \in \Omega_j} (S_{ij} - \mathbf{b}_i^\top \bar{\mathbf{d}}_j^\top \mathbf{d}_{j_k}^\top) b_{ik} + \beta y_{jk}. \end{aligned} \quad (4)$$

W-subproblem

Similar to the D-subproblem, we parallelly update \mathbf{W} according to

$$\underset{\mathbf{w}_t \in \{\pm 1\}^r}{\text{argmin}} \mathbf{w}_t^\top \left(\sum_{i \in \Gamma_t} \mathbf{b}_i \mathbf{b}_i^\top \right) \mathbf{w}_t - 2 \left(\sum_{i \in \Gamma_t} T_{it} \mathbf{b}_i^\top \right) \mathbf{w}_t - 2\gamma \mathbf{z}_t^\top \mathbf{w}_t.$$

We use DCD to update w_t based on the following rule:

$$\begin{aligned} w_{tk} &\leftarrow \text{sgn}(K(\check{w}_{tk}, w_{tk})) \\ \check{w}_{tk} &= \sum_{i \in \Gamma_t} (T_{it} - \mathbf{b}_i^\top \bar{\mathbf{w}}_t^\top \mathbf{w}_{tk}^\top) b_{ik} + \gamma z_{tk}. \end{aligned} \quad (5)$$

X-subproblem

In this subproblem, we aim to optimize \mathbf{X} with fixed $\mathbf{B}, \mathbf{D}, \mathbf{W}, \mathbf{Y}, \mathbf{Z}$. The optimization for \mathbf{X} is:

$$\underset{\mathbf{X}}{\text{argmax}} \text{tr}(\mathbf{B}^\top \mathbf{X}), \text{ s.t. } \mathbf{X} \mathbf{1}_m = 0; \mathbf{X} \mathbf{X}^\top = m \mathbf{I}_r.$$

Inspired by [Liu *et al.*, 2014], we denote a matrix $\mathbf{J} = \mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top$, by Singular Value Decomposition (SVD), we have $\bar{\mathbf{B}} = \mathbf{B} \mathbf{J} = \mathbf{P}_B \Sigma_B \mathbf{Q}_B^\top$, where $\mathbf{P}_B \in \mathbf{R}^{h \times h'}$ and $\mathbf{Q}_B \in \mathbf{R}^{m \times h'}$ are the matrix of the left- and right- singular vectors corresponding to the first $h' (\leq h)$ singular values of matrix Σ , respectively. In fact, we perform SVD for the small $h \times h$ matrix $\bar{\mathbf{B}} \bar{\mathbf{B}}^\top = [\mathbf{P}_B \hat{\mathbf{P}}_B] \text{Diag}[\Sigma_B^2 \mathbf{0}] [\mathbf{P}_B \hat{\mathbf{P}}_B]^\top$, where $\hat{\mathbf{P}}_B$ are the eigenvectors of the zero eigenvalues. By the definition of SVD, we have $\mathbf{Q}_B = \bar{\mathbf{B}}^\top \mathbf{P}_B \Sigma_B^{-1}$. In addition, we obtain additional $\hat{\mathbf{Q}}_B \in \mathbf{R}^{m \times (h-h')}$ by Gram-Schmidt orthogonalization based on $[\mathbf{Q}_B \mathbf{1}]$ in which \mathbf{B} satisfies $\mathbf{B} \mathbf{1} = 0$. Thus, we have $\hat{\mathbf{Q}}_B^\top \mathbf{1} = 0$. Meanwhile, since \mathbf{Q}_B and $\bar{\mathbf{B}}$ have the same row space, and $\bar{\mathbf{B}} \mathbf{1} = 0$, so we have $\mathbf{Q}_B^\top \mathbf{1} = 0$. We can get $[\mathbf{Q}_B \hat{\mathbf{Q}}_B]^\top \mathbf{1} = 0$, which implies $\mathbf{X} \mathbf{1} = 0$. At the same time, we can get $\mathbf{X} \mathbf{X}^\top = m [\mathbf{P}_B \hat{\mathbf{P}}_B] [\mathbf{Q}_B \hat{\mathbf{Q}}_B]^\top [\mathbf{Q}_B \hat{\mathbf{Q}}_B] [\mathbf{P}_B \hat{\mathbf{P}}_B]^\top = m \mathbf{I}$. We can prove that such \mathbf{X} is the optimal solution, which can be referred to [Zhang *et al.*, 2016]. So we can get the closed-form solution of X-subproblem is

$$\mathbf{X} \leftarrow \sqrt{m} [\mathbf{P}_B \hat{\mathbf{P}}_B] [\mathbf{Q}_B \hat{\mathbf{Q}}_B]^\top. \quad (6)$$

Y-subproblem

In this subproblem, we aim to optimize \mathbf{Y} with fixed $\mathbf{B}, \mathbf{D}, \mathbf{W}, \mathbf{X}, \mathbf{Z}$. The objective of Y-subproblem is:

$$\underset{\mathbf{Y}}{\text{argmax}} \text{tr}(\mathbf{D}^\top \mathbf{Y}), \text{ s.t. } \mathbf{Y} \mathbf{1}_n = 0; \mathbf{Y} \mathbf{Y}^\top = n \mathbf{I}_r.$$

Similar with X-subproblem, we can derive a closed-form solution as follows:

$$\mathbf{Y} \leftarrow \sqrt{n} [\mathbf{P}_D \hat{\mathbf{P}}_D] [\mathbf{Q}_D \hat{\mathbf{Q}}_D]^\top. \quad (7)$$

Algorithm 1 Discrete Trust-aware Matrix Factorization

Input: S: ratings matrix; Γ : trusts matrix; r: code length

Output: user,item binary code: \mathbf{B}, \mathbf{D}

```

1: Initialization:  $\mathbf{B}, \mathbf{W}, \mathbf{X}, \mathbf{Z} \in \mathbf{R}^{r \times m}$  and  $\mathbf{D}, \mathbf{Y} \in \mathbf{R}^{r \times n}$ 
2: while not converge do
3:   for  $i \in \{1, \dots, m\}$  do
4:     repeat
5:       use Eq.(3) to update  $B_i$  bit by bit (r bits in total)
6:     until bit converge
7:   end for
8:   for  $j \in \{1, \dots, n\}$  do
9:     repeat
10:      use Eq.(4) to update  $D_j$  bit by bit (r bits in total)
11:    until bit converge
12:   end for
13:   for  $k \in \{1, \dots, m\}$  do
14:     repeat
15:      use Eq.(5) to update  $W_k$  bit by bit (r bits in total)
16:    until bit converge
17:   end for
18:   update X by Eq.(6)
19:   update Y by Eq.(7)
20:   update Z by Eq.(8)
21: end while
22: return  $\mathbf{B}, \mathbf{D}$  to evaluate

```

Z-subproblem

In this subproblem, we aim to optimize \mathbf{Z} with fixed $\mathbf{B}, \mathbf{D}, \mathbf{W}, \mathbf{X}, \mathbf{Y}$. The goal of Z-subproblem is:

$$\underset{\mathbf{Z}}{\text{argmax}} \text{tr}(\mathbf{W}^\top \mathbf{Z}), \text{ s.t. } \mathbf{Z} \mathbf{1}_m = 0; \mathbf{Z} \mathbf{Z}^\top = m \mathbf{I}_r.$$

Similar with X-subproblem, we can get the closed-form solution of Z-subproblem as follows:

$$\mathbf{Z} \leftarrow \sqrt{m} [\mathbf{P}_W \hat{\mathbf{P}}_W] [\mathbf{Q}_W \hat{\mathbf{Q}}_W]^\top. \quad (8)$$

3.4 Better Initialization of Discrete Problems

Since our DTMF model deals with a mixed-integer non-convex optimization problem, better initialization usually contributes to achieving faster convergence and better local optimal solutions, as pointed out by [Zhang *et al.*, 2016; Lian *et al.*, 2017]). This is done by solving the following relaxation optimization problem, i.e., by replacing real-valued variables $\mathbf{U}, \mathbf{V}, \mathbf{F}$ with binary ones $\mathbf{B}, \mathbf{D}, \mathbf{W}$, respectively.

$$\begin{aligned} \underset{\mathbf{U}, \mathbf{V}, \mathbf{F}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}}{\text{argmin}} & \sum_{i,j \in \Omega} (S_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 + \lambda \sum_{i,t \in \Gamma} (T_{it} - \mathbf{u}_i^\top \mathbf{f}_t)^2 \\ & - 2\alpha \text{tr}(\mathbf{U}^\top \mathbf{X}) - 2\beta \text{tr}(\mathbf{V}^\top \mathbf{Y}) - 2\gamma \text{tr}(\mathbf{F}^\top \mathbf{Z}) \\ & + \alpha \|\mathbf{U}\|_F^2 + \beta \|\mathbf{V}\|_F^2 + \gamma \|\mathbf{F}\|_F^2 \\ \text{s.t. } & \mathbf{X} \mathbf{1}_m = 0; \mathbf{Y} \mathbf{1}_n = 0; \mathbf{Z} \mathbf{1}_m = 0; \\ & \mathbf{X} \mathbf{X}^\top = m \mathbf{I}_r; \mathbf{Y} \mathbf{Y}^\top = n \mathbf{I}_r; \mathbf{Z} \mathbf{Z}^\top = m \mathbf{I}_r. \end{aligned}$$

We can use gradient descent algorithm to solve U-, V-, and W-subproblem, and take the SVD method to solve X-, Y-, and Z-subproblem. Assuming the solutions are denoted by

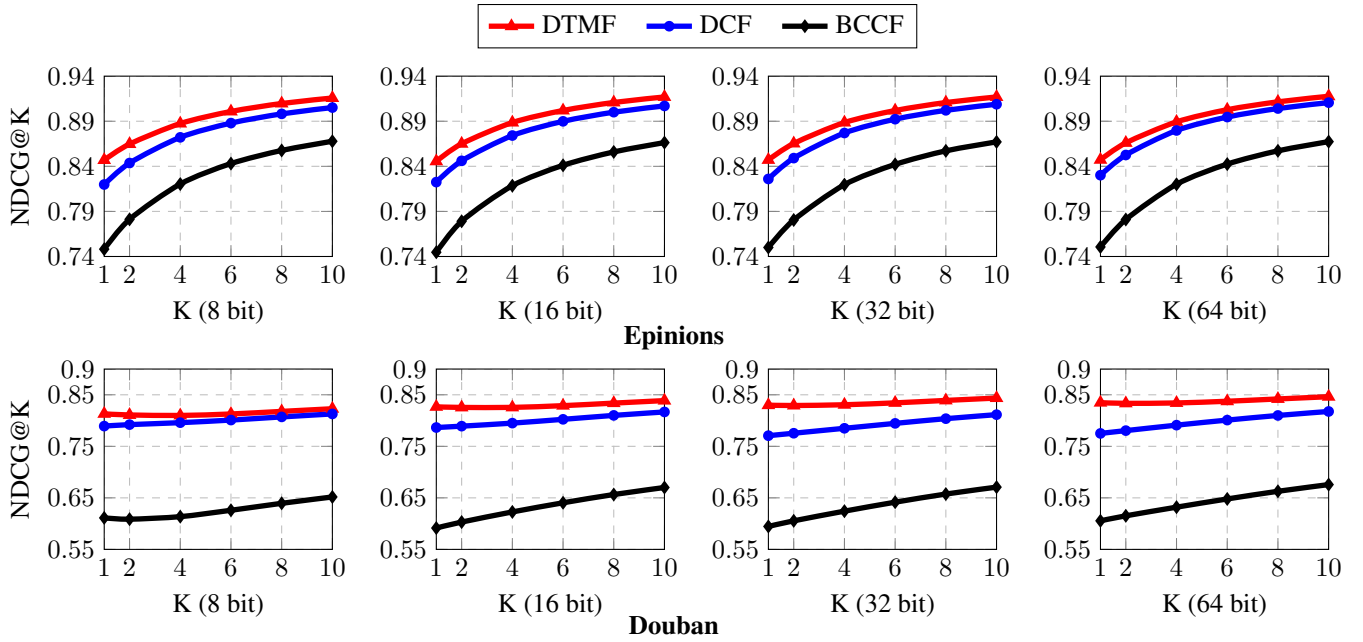


Figure 1: Performance (vs. Discrete) of NDCG@K on Epinions and Douban Datasets, respectively

U^* , V^* , F^* , X^* , Y^* , Z^* , we can then initialize the discrete objective function as follows:

$$\begin{aligned} B &\leftarrow \text{sgn}(U^*), D \leftarrow \text{sgn}(V^*), W \leftarrow \text{sgn}(F^*), \\ X &\leftarrow X^*, Y \leftarrow Y^*, Z \leftarrow Z^*. \end{aligned}$$

4 Experiments

4.1 Experimental Settings

Datasets and Baselines

Two real-world datasets are used in our experiments, namely Epinions² and Douban³. Both datasets consist of social networks and rating data. Note that Epinions supports the concept of directed trust, while Douban takes bi-directional friendship as social connections. The statistics of the two datasets are presented in Table 1.

We compare with the following approaches:

- **BCCF** [Zhou and Zha, 2012]: It is a two-stage binarized collaborative filtering method with a relaxation stage and a quantization stage.
- **DCF** [Zhang *et al.*, 2016]: It is the first binarized CF method that can jointly optimize the binary codes.
- **TrustMF** [Yang *et al.*, 2013]: It is a real-valued matrix factorization method based upon both user-item ratings and trust information.

Evaluation method

In our experiments, for each user, we randomly selected 50% as training data and the rest as test data. We perform each experiment five times and report the average results as the final performance. The Normalized Discounted Cumulative Gain

Dataset	User	Item	Rating	Trust
Epinions	49,290	139,738	664,824	487,183
Douban	129,490	58,541	16,830,839	1,692,952

Table 1: Statistics of the Datasets

(NDCG) is used as our evaluation metric to estimate the quality of recommendation accuracy [Valizadegan *et al.*, 2009]. Generally, higher NDCG values indicate better ranking accuracy.

Parameter settings

The parameters of all the methods are either determined by empirical study or suggested by the original paper. Specifically, for BCCF, we tune hyper-parameters λ within $[10^{-4}, \dots, 10^{-2}]$. The hyper-parameters α and β of DCF are tuned with $[10^{-4}, \dots, 10^2]$. For TrustMF, we adopt the parameter settings recommended by the authors: $\lambda = 0.001$ and $\lambda_T = 1$. For DTMF proposed in this paper, we search α, β, γ , and λ from $[10^{-4}, \dots, 10^3]$.

4.2 Results and Analysis

Compared to discrete models

In Figure 1, we give the NDCG@1 to NDCG@10 values for all the compare methods on two datasets, and observe that:

- Compare of BCCF: The performance of BCCF is much lower than that of both DCF and DTMF, which are discrete methods. This is because BCCF is a two-stage discretization method, which brings a large quantization loss. Both DCF and DTMF are joint optimization discrete methods, and thus the characteristics of real-valued and discrete-valued transformation are well preserved.
- Compare of DCF: The results show that our approach DTMF is consistently better than DCF. We attribute it to the underlying assumption that users are socially connected and mutually influenced with each other. Social

²http://www.trustlet.org/wiki/Epinions_dataset

³<https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban>

ties have a positive impact on recommendation performance. In contrast, DCF assumes that all users are independent identically distributed, which thus ignores the relationships among users.

- Compare of datasets: By comparing the Douban with Epinions dataset, we find that our model DTMF is more improved than DCF and BCCF in the Douban dataset. It may be explained by the fact that Douban provides more social information than Epinions (see Table 1).

Compared to real-valued model

The purpose of this part is to investigate if our discrete DTMF performs closely to the real-valued model, i.e., TrustMF. The results are shown in Figure 2. It can be seen that DTMF reaches very close NDCG values to TrustMF on Epinions, whereas it beats TrustMF on Douban in the case of 32 bits. Note that other results with different bits basically follow the similar trend, and thus we omit them for space saving. The results may be explained in two aspects. First, Douban provides more social information than Epinions (see Table 1), leading to more exposure of user influence on each other. Second, Douban adopts the concept of friendship (bi-directional) while Epinions support the connections of trust (directed). By definition, trust is generally believed a stronger social tie than friendship. Thus, trust-aware real-valued model TrustMF performs worse in Douban, while discrete DTMF may be more applicable in different social datasets because of discretion operations. As a conclusion, discrete trust-aware model achieves comparable or even better performance than the real-valued one, and demonstrates better application to different social ties.

Efficiency study

Recommendation efficiency is an important aspect to study, and the results regarding to data size and code length are depicted in Figure 3. The experiments are executed on Douban due to its greater size. Firstly, we fix the code length as 8 bits, and gradually increase the size of training data. Secondly, we fix the training ratio (50% for training), and then tune the value of code length. Consistently, in both cases, the time required per iteration will be linearly increased, indicating the potential to be applied in large-scale datasets. Besides, we can save the learned model in smaller memory after model training. Note that our results are consistent with the conclusion of existing discrete collaborative filtering models [Zhang *et al.*, 2016; Liu *et al.*, 2018], that is, the recommendation efficiency is much better than that of real-valued counterparts.

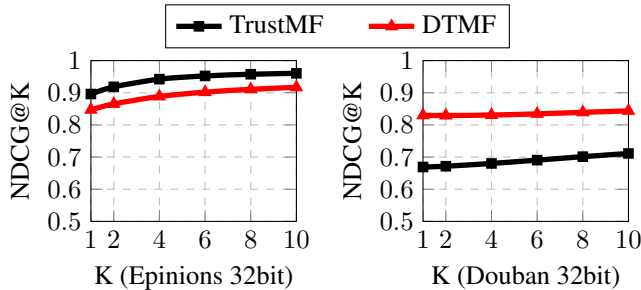


Figure 2: Performance comparison with TrustMF in NDCG@K

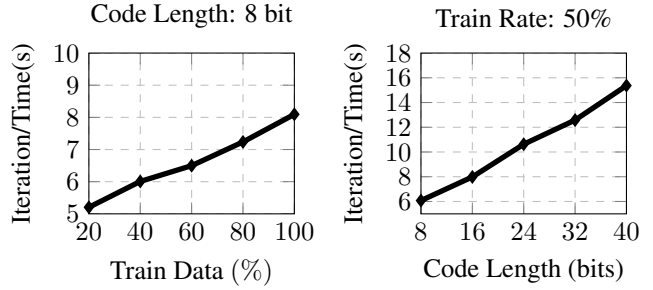


Figure 3: Efficiency w.r.t. data size and code length on Douban

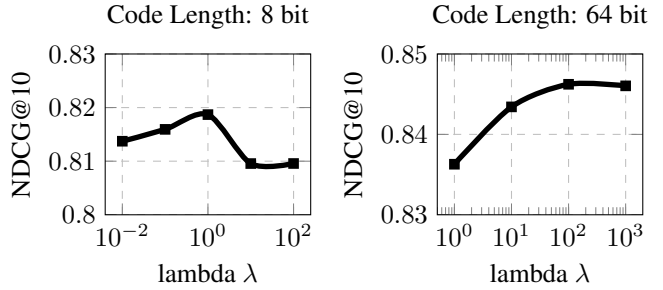


Figure 4: Impact of Hyper-parameter λ on Douban

As a result, discrete models can reach comparable performance and better efficiency, and thus even suitable for search problems.

Impact of hyper-parameter

Parameter λ controls the importance of trust modeling for the overall performance, and the results are illustrated in Figure 4. It can be observed that in general tuning values of parameter λ will not greatly change the overall performance. The NDCG values vary in a relatively small range. Nevertheless, a proper setting of parameter λ can help reach the best performance. The optimal setting is also a dataset specific.

5 Conclusions

In this paper, we proposed a novel discrete trust-aware matrix factorization model called DTMF. It took advantages of both hash technique (for better efficiency) and social influence (for better accuracy) to decompose user-item ratings, which mapped both users and items into a joint hamming space. We applied an efficient discrete optimization algorithm to resolve each subproblem of our model alternatively. The results on two real datasets demonstrated that our approach can obtain better efficiency yet preserve comparable (or even better) accuracy in comparison with a number of counterparts.

Acknowledgments

This work is supported by the JD Grapevine Plan and Fundamental Research Funds for the Central Universities under Grant No. N181705007, by the National Natural Science Foundation of China under Grants No. 61772125, No. 61702084, No. 61702090 and No. 61772570, by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant No. 2018B030306025, Pearl River S&T Nova Program of Guangzhou under Grant No. 201806010056, and by the Fundamental Research Funds for the Central Universities No. N171602003.

References

- [Abhinandan *et al.*, 2007] Das Abhinandan, Datar Mayur, Garg Ashutosh, and Rajaram Shyam. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, May 2007.
- [Guo *et al.*, 2015] Guibing Guo, Jie Zhang, and Yorke-Smith Neil. Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *29th AAAI Conference on Artificial Intelligence*, pages 123–129. AAAI, January 2015.
- [Lian *et al.*, 2017] Defu Lian, Rui Liu, Yong Ge, Kai Zheng, Xing Xie, and Longbing Cao. Discrete content-aware matrix factorization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 325–334. ACM, August 2017.
- [Liu *et al.*, 2014] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang. Discrete graph hashing. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3419–3427. Curran Associates, Inc., December 2014.
- [Liu *et al.*, 2018] Han Liu, Xiangnan He, Fuli Feng, Liqiang Nie, Rui Liu, and Hanwang Zhang. Discrete factorization machines for fast feature-based recommendation. In *Proceedings of the 27rd International joint Conference on Artificial Intelligence*, pages 3449–3455. AAAI, August 2018.
- [Liu *et al.*, 2019] Chenghao Liu, Xin Wang, Tao Lu, Wenwu Zhu, Jianling Sun, and Steven Hoi. Discrete social recommendation. AAAI, 2019.
- [Ma *et al.*, 2008] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 931–940. ACM, October 2008.
- [Ma *et al.*, 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 287–296. ACM, February 2011.
- [Mohsen and Martin, 2010] Jamali Mohsen and Ester Martin. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 135–142. ACM, September 2010.
- [Valizadegan *et al.*, 2009] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *Advances in Neural Information Processing Systems 22*, pages 1883–1891. Curran Associates, Inc., 2009.
- [Wang *et al.*, 2016] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to hash for indexing big data—a survey. *Proceedings of the IEEE*, 104(1):34–57, December 2016.
- [Yang *et al.*, 2013] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. Social collaborative filtering by trust. In *Proceedings of the 23rd International joint Conference on Artificial Intelligence*, pages 2747–2753. AAAI, August 2013.
- [Zhang *et al.*, 2014] Zhiwei Zhang, Qifan Wang, Lingyun Ruan, and Luo Si. Preference preserving hashing for efficient recommendation. In *Proceedings of the 37th International ACM SIGIR Conference on Research Development in Information Retrieval*, pages 183–192. ACM, July 2014.
- [Zhang *et al.*, 2016] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. Discrete collaborative filtering. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334. ACM, July 2016.
- [Zhang *et al.*, 2017] Yan Zhang, Defu Lian, and Guowu Yang. Discrete personalized ranking for fast collaborative filtering from implicit feedback. In *31th AAAI Conference on Artificial Intelligence*, pages 1669–1675. AAAI, February 2017.
- [Zhou and Zha, 2012] Ke Zhou and Hongyuan Zha. Learning binary codes for collaborative filtering. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 498–506. ACM, August 2012.