Neurocomputing 507 (2022) 145–156

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Binary multi-modal matrix factorization for fast item cold-start recommendation

Chengmei Peng [a], Lei Zhu [a], Yang Xu [a,*], Yaping Li [b], Lei Guo [a]

[a] School of Information Science and Engineering, Shandong Normal University, China
[b] School of Information Engineering, Shandong Management University, China

ARTICLE INFO

Article history:
Received 2 August 2021
Revised 28 May 2022
Accepted 3 August 2022
Available online 6 August 2022
Communicated by Zidong Wang

Keywords:
Hashing
Binary
Multi-modal
Matrix factorizaton
Item cold-start recommendation

ABSTRACT

Hashing technology can support large-scale recommendation very effectively due to its advantages of low storage cost and high recommendation efficiency. However, existing hashing-based recommendation methods often suffer from item cold-start problem. This is because they simply consider the user-item interaction and the single content information of the items, but the full interaction history is not always available and the single auxiliary information may be missing. To solve this issue, in this paper we propose a *Binary Multi-modal Matrix Factorization* (BMMF) method. First, we propose an efficient consensus multi-modal mapping to transform the heterogeneous multi-modal features to the unified factors by exploiting the complementarity of multiple modalities. Then, binary matrix factorization is simultaneously performed on the multi-modal features of the items and past user preferences to learn the compact binary codes of the users/items in a common Hamming space. In addition, inspired by the observation that similar instances often have similar binary codes within a short Hamming distance, we formulate a semantic structure regularization term to preserve the similarities of the items during the binary embedding process. Finally, we develop an effective Discrete Coordinate Descent (DCD) approach to tackle the formulated discrete hash optimization problem directly. Experiments on three publicly available real-world datasets demonstrate the superiority of the proposed method against the state-of-the-art methods. Our source codes and testing datasets are available at https://github.com/pcm1217/BMMF.

1. Introduction

Recommender systems have been a direct way to prevent consumers from making excessive choices [1]. Personalized recommendation technology has received wide attention from researchers, and it has been deeply investigated by many large Internet companies (such as Amazon[1] and Alibaba[2]) because it can recommend relevant items to users and increase the users' stickiness and loyalty.

Collaborative Filtering (CF) based on Matrix factorization [2,3] is one of the most widely adopted techniques in recommender systems. It factorizes an $m \times n$ user-item rating matrix to project m users and n items into an r-dimensional latent feature space. The inner product of the latent features is used to predict the user's preference for items. Actually, the top-K item recommendation for a user can be transformed into the problem of producing a ranking list of responses (item) to the query (user) [4]. However, with the explosive increase in users and items, recommender systems may suffer from serious efficiency problems.

Recently, hashing technology [5] has been introduced to large-scale recommendation systems due to its desirable advantages of low storage cost and high recommendation efficiency. The core of hashing-based recommendation methods is to encode real-valued latent features into compact binary codes, and then measure the similarities between the users and items with a Hamming distance. In this case, the similarity calculation by the inner product can be efficiently performed by bit-wise operations. One can even find approximate top-K preferred items with a sub-linear or constant time complexity [6,7]. However, learning compact binary codes with discrete constraints is an intractable NP-hard problem [8]. To address this problem, the two-stage learning procedure [9] is proposed, which first solves the relaxed optimization problem by discarding the binary constraints and then performs a direct binary quantization. However, as indicated by [10], those two-stage learning approaches will result in a significant quantization loss. Therefore, several discrete hashing methods have been developed to directly learn binary codes from matrix factorization

* Corresponding author.
E-mail address: zzmylq@gmail.com (Y. Xu).

1 https://www.amazon.com/.
2 https://www.1688.com/.

https://doi.org/10.1016/j.neucom.2022.08.013
0925-2312/

without discarding the binary constraints [11–14]. Although promising progress has been made, there are two important problems that have not yet been solved well. First, the item cold-start problem is challenging because most hashing-based recommendation methods mainly rely on user-item interactions and single auxiliary features, while ignoring the multi-modal features of the items. Intuitively, when the recommender systems have no interaction history with the items or lack specific auxiliary information, the cold-start problem cannot be handled well. Second, the similarity between items is not maintained when learning hash codes. Existing hashing-based recommendation methods ignore the fact that items with semantical similarities should also have short Hamming distances in the binary codes.

To alleviate these limitations, we propose a *Binary Multi-modal Matrix Factorization* (BMMF) method in this paper. First, we develop an efficient consensus multi-modal mapping to transform the multi-modal features of the items to unified factors by exploiting the complementarity of the multi-modal features, and preserving the semantic correlations in multiple modalities. Then, binary matrix factorization is simultaneously performed on the multi-modal features of the items and the users' past preferences to learn the compact hash codes of the users/items in a common Hamming space. Finally, to effectively preserve the similarities of the items, we formulate a semantic structure regularization term to enforce similar items to share similar binary codes during the binary embedding process. The main contributions of the BMMF method are outlined as follows:

- We propose an efficient item recommendation framework based on binary multi-modal matrix factorization, which projects the multi-modal features of the items and preserves the users' past preferences into a common Hamming space.
- We develop an efficient consensus multi-modal mapping to transform multi-modal features of the items to unified factors while exploiting the complementarity of multiple modalities. In addition, the semantic structure regularization term is formulated to enhance the item similarity preservation during the binary embedding process.
- We develop an effective Discrete Coordinate Descent (DCD) approach to tackle the formulated discrete hash optimization problem. Experiments on three publicly available recommendation datasets demonstrate that the proposed method achieves a state-of-the-art performance.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 gives details of the proposed method. Section 4 presents the experimental configurations and the results. Finally, Section 5 concludes the paper.

2. Related work

2.1. Cold-start recommendation

In recommender systems, recommend new items without any user historical feedback is a demanding and challenging process, which leads to the cold-start recommendation problem. Traditional methods incorporate CF embeddings and content features to address the cold-start recommendation problem [15,16]. In addition, several cross-domain recommendation methods [17–19] have been proposed to alleviate the cold-start problem by transferring the learned latent features of overlapped objects from the auxiliary domain to the target domain. However, they are only applicable to specific recommendation scenarios. Some efforts [20–22] introduce meta-learning [23] into the recommender systems to solve the cold-start problem by a fast adaptation with only a few samples for users and items. The key idea of meta learning-based recommendation methods [24] is to learn a personalized recommendation with globally initialized parameters, which are further locally updated to predict a new user's preference. However, most of the methods focus on producing personalized parameter initialization for new users instead of new items.

Moreover, the aforementioned methods use the inner product to calculate the recommendation results. Thus, they will suffer from serious efficiency problems in large-scale recommendation scenarios.

2.2. Hashing-based recommendation

Hashing is a promising technique to solve the efficiency problem of large-scale recommender systems and a detailed introduction can be found in [5]. Hashing-based recommendation can be organized into two categories: two-stage hashing and discrete hashing [25]. Two-stage hashing methods consist of relaxed optimization stage and binary quantization stage. However, such two-stage hashing approaches suffer from a significant quantization loss [10].

Discrete hashing methods can reduce large quantization errors by directly solving the discrete optimization problem. Discrete Collaborative Filtering (DCF) [10] is proposed to directly learn the binary codes of the users and items from explicit feedback. It adds balanced and decorrelated constraints to discrete matrix factorization while enforcing the discreteness of the generated hash codes. Inspired by DCF, Discrete Personalized Ranking (DPR) [12] is proposed for personalized ranking from implicit feedback (in contrast to explicit feedback in the DCF case). It optimizes a ranking objective the Area Under ROC curve (AUC) to learn hash codes similar to DCF. However, DCF and DPR only use rating data and easily suffer from the cold-start recommendation problem. Following the basic idea, Liu et al. [14] propose Discrete Factorization Machine (DFM) for general feature-based recommendation. DFM can learn binary embedding for each feature and model the pairwise interactions between features. Lian et al. [11] develop a discrete optimization method called Discrete Content-aware Matrix Factorization (DCMF), which can address both explicit and implicit feedback. It learns binary codes with respect to the different side information of the users and items. Discrete Deep Learning (DDL) [13] integrates a Deep Belief Network (DBN) [26] and Collaborative Filtering (CF) framework to learn hash codes based on ratings and auxiliary information. Deep Pairwise Hashing (DPH) [27] extends DDL with a ranking-based hashing framework, it chooses Denoising Autoencoder (DAE) [28] instead of the DBN embedded in DDL to learn robust item representations. Discrete Matrix Factorization (DMF) [29] is an extension of DCMF [11]. It develops a Block Coordinate Descent (BCD) [30] based on the semidefinite relaxation of binary quadratic programming to tackle the discrete optimization problem. Hansen et al. [31] present a content-aware neural hashing-based collaborative filtering approach (NeuHash-CF), which adopts a variational autoencoder architecture, including two joint hashing components to obtain the binary codes of the users and items in different settings. Product Quantized Collaborative Filtering (pQCF) [25] takes rating data as input and unifies the quantization and the learning of latent factors into a joint framework, which is based on the inner product space rather than the Euclidean space. However, the above recommendation methods mainly depend on user-item interactions and single specific auxiliary features. When specific auxiliary information is missing, these methods fail to provide meaningful recommendations for new users and items. In addition, these methods cannot take full advantage of the rich multi-modal auxiliary information. To this end, Multi-Feature Discrete Collaborative Filtering (MFDCF) [32] is proposed as a unified multi-feature discrete collaborative filtering model that maps the multiple content features of the users into the binary codes with the learned hash project matrix. In the optimization process, the hash codes can be learned with a simple $\mathrm{sgn}(\cdot)$[3] operation. However, the direct mapping method used by MFDCF results in a larger quantization loss than using the DCD method. Note that all these approaches learn the hash codes based on the user-item interactions and auxiliary information, but they fail to consider the similarity preservation of the items.

The differences between the existing methods and ours are as follows. First, our proposed model employs the complementarity of the multi-modal features of the items, which can effectively solve the item cold-start recommendation problem. Second, the proposed model contains a unified learning framework that preserves the intrinsic user-item relevance and item-item relations into binary codes. Finally, the framework employs balance and decorrelated constraints to generate compact but informative binary codes.

3. The proposed method

Notations. We utilize bold uppercase and lowercase letters to represent matrices and vectors, respectively, and adopt non-bold letters for scalars. Particularly, $\mathbf{a}_i$ is the i-th row vector of matrix $\mathbf{A}, a_{ij}$ is the element at the i-th row and j-th column of $\mathbf{A}$. $\|\mathbf{A}\|_F$ and $tr(\mathbf{A})$ are the Frobenius norm and trace of matrix $\mathbf{A}$, respectively. In addition, $\mathrm{sgn}(\cdot):\mathbb{R}\rightarrow\{\pm 1\}$ is the round-off function. The main notations and their corresponding descriptions are summarized in Table 1.

Table 1
Notations and descriptions.

Notation	Description
$\mathbf{H}^l$	feature matrix of the l-th modality data
$\varphi\left(\mathbf{h}^l\right)$	nonlinear embedding of the l-th modality feature
$\varphi(\mathbf{H})$	consensus multi-modal feature matrix
$\mathbf{R}$	the user-item rating matrix
$\mathbf{S}$	affinity matrix
$\mathbf{B}$	hash codes of users
$\mathbf{C}$	hash codes of the mapping matrix of L modalities
$\mathbf{D}$	hash codes of items
$\mathbf{U}$	continuous features of users
$\mathbf{V}$	continuous features of items
$\mathbf{W}$	continuous features of the mapping matrix of L modalities
$\mathbf{X}$	intermediate real matrix of $\mathbf{B}$
$\mathbf{Y}$	intermediate real matrix of $\mathbf{D}$
$\mathbf{Z}$	intermediate real matrix of $\mathbf{C}$
m	the number of users
n	the number of items
p	the number of anchors
L	the number of modalities
r	the length of hash code

3.1. Problem formulation

Assume there are m users and n items, then the rating matrix $\mathbf{R}$ is of size $n\times m$, where r_{ij} represents the preference of user j for item i. Suppose that $\mathbf{O}=\{\mathbf{o}_i\}_{i=1}^n$ is the item training dataset, which contains n items represented with L different modality features (e.g., movie information denoted by author, genre, title). The l-th modality feature of items is denoted as $\mathbf{H}^l=\left[\mathbf{h}_1^l,\mathbf{h}_2^l,\ldots,\mathbf{h}_n^l\right]^{\mathrm{T}}\in\mathbb{R}^{n\times d_l}$ $(l=1,2\ldots,L)$, where d_l is the corresponding feature dimension. Our proposed method aims at learning hash codes $\mathbf{B}\in\{\pm 1\}^{r\times m}$ for users and $\mathbf{D}\in\{\pm 1\}^{r\times n}$ for items to represent their latent factors in the offline training stage, where r is the hash code length. The hash codes of the cold-start items are obtained by efficient online hashing, and the recommendation results are efficiently generated by calculating the Hamming distance between the hash codes in the online recommendation stage. The basic framework of the proposed BMMF is illustrated in Fig. 1.

3.2. Consensus multi-modal feature mapping

Given a training dataset $\mathbf{O}=\{\mathbf{o}_i\}_{i=1}^n$. Since the feature dimensions of multiple modalities may be different, we construct a consensus multi-modal feature mapping that nonlinearly transforms the uni-modal feature $\mathbf{h}_i^l$ into a unified embedding $\varphi\left(\mathbf{h}_i^l\right)$. That is,

$$\varphi\left(\mathbf{h}_i^l\right)=\left[\exp\left(\frac{\left\|\mathbf{h}_i^l-\mathbf{a}_1^l\right\|_F^2}{2\sigma_l^2}\right),\ldots,\exp\left(\frac{\left\|\mathbf{h}_i^l-\mathbf{a}_p^l\right\|_F^2}{2\sigma_l^2}\right)\right],\quad(1)$$

where $\left\{\mathbf{a}_1^l,\mathbf{a}_2^l,\ldots,\mathbf{a}_p^l\right\}$ are p anchors randomly selected from the l-th modality in the training set, and σ_l is the Gaussian kernel bandwidth. The modality-specific sample relationship can be preserved by $\varphi\left(\mathbf{h}_i^l\right)\in\mathbb{R}^{1\times p}$ which characterizes the correlations between samples and certain anchors. Specifically, we suggest an intermediate consensus multi-modal feature $\varphi(\mathbf{h}_i)$ to exploit the complementarity of multi-modal features, which is formulated as:

$$\varphi(\mathbf{h}_i)=\left[\mu_1^{\frac{1}{2}}\varphi\left(\mathbf{h}_i^1\right),\ldots,\mu_L^{\frac{1}{2}}\varphi\left(\mathbf{h}_i^L\right)\right]\in\mathbb{R}^{1\times Lp},\quad(2)$$

where μ_i is the weight to measure the contribution of the i-th modality, the above formula is composed of L unique embedded features. We can derive the consensus multi-modal feature matrix $\varphi(\mathbf{H})$ as:

$$\varphi(\mathbf{H})=[\varphi(\mathbf{h}_1);\ldots;\varphi(\mathbf{h}_n)]\in\mathbb{R}^{n\times Lp}.\quad(3)$$

The heterogeneous modality gap among the different modalities may damage hash code learning. Hence, directly using $\varphi(\mathbf{H})$ to learn hash codes may result in a sub-optimal performance. To address this problem, we design a multi-modal feature fusion method to obtain the continuous features of items $\mathbf{V}=[\mathbf{v}_1,\ldots,\mathbf{v}_n]\in\mathbb{R}^{r\times n}$ from $\varphi(\mathbf{H})$, where r is the feature dimension (hash code length). The formula is:

$$\begin{aligned}&\min_{\mathbf{V},\mathbf{W},\boldsymbol{\mu}}\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij}-\mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j\right)^2,\\&\text{s.t.}\quad\boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L=1,\quad\boldsymbol{\mu}\geqslant\mathbf{0},\end{aligned}\quad(4)$$

where $\varphi(\mathbf{H})_{ij}$ is the observed multi-modal feature representation, whose index set is $\Omega,\boldsymbol{\mu}=[\mu_1,\mu_2,\ldots,\mu_L]$ is the weight vector, $\mathbf{1}_L$ is an all-one column vector with length L, and $\mathbf{W}=[\mathbf{w}_1,\ldots,\mathbf{w}_{Lp}]\in\mathbb{R}^{r\times Lp}$ is the mapping matrix of L modalities to reduce the inter-modal redundancy. The inner product $\mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j$ between the item feature $\mathbf{v}_i$ and the multi-modal feature mapping $\mathbf{w}_j$ can be used to approximate an observed multi-modal feature representation $\varphi(\mathbf{H})_{ij}$.

3.3. Binary multi-modal matrix factorization

The main goal of our work is to generate consensus feature representations of the users and items by exploiting the multi-modal auxiliary information and user-item interaction [33]. In the previous section, we fuse the multi-modal information into V by nonlinear mapping. Next, we adopt matrix factorization to extract the interaction features and save them in V. We define our objective function as shown in Eq. (5), given by:

[3] $\mathrm{sgn}(\cdot):\mathbb{R}\rightarrow\{\pm 1\}$ is the round-off function.

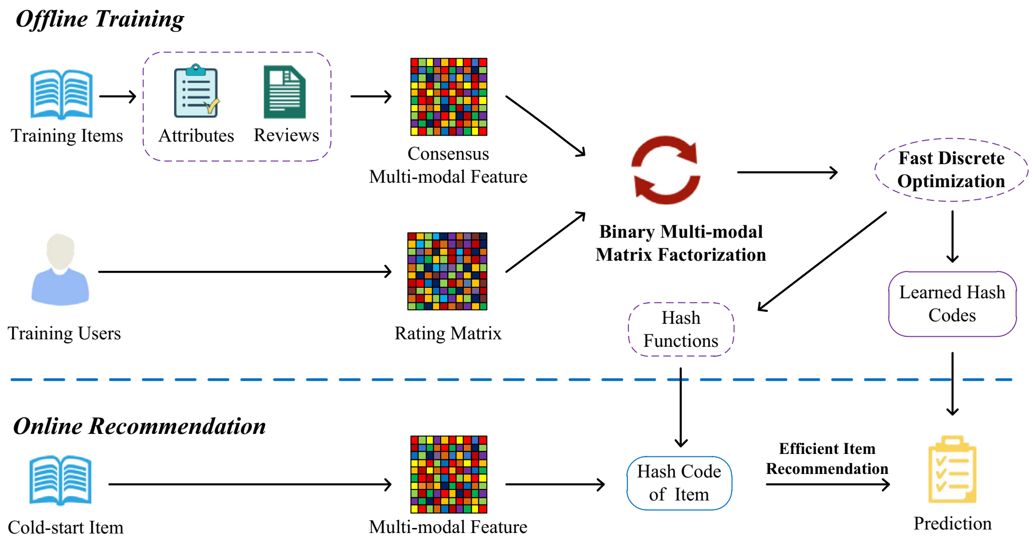

Fig. 1. The basic framework of the proposed BMMF, which consists of two stage: offline training and online recommendation. The main task of the offline training stage is to learn the hash functions and generate the hash codes for the users and items by employing the complementarity of the multi-modal features. In the online recommendation stage, the heterogeneous multi-modal features of the cold-start items are first extracted and projected into intermediate consensus feature representations. Then, the binary codes can be quickly generated by the proposed binary multi-modal matrix factorization method and the learned hash functions.

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{W},\boldsymbol{\mu}}\ & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij}-\mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j\right)^2+(1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih}-\mathbf{v}_i^{\mathrm{T}}\mathbf{u}_h\right)^2\\ & +\alpha\left(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2+\|\mathbf{W}\|_F^2\right),\\ & \text{s.t.}\,\boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L=1,\boldsymbol{\mu}\geqslant\mathbf{0},\end{aligned}\tag{5}$$

where r_{ih} is the observed rating, Ψ denotes the index set, $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_m]\in\mathbb{R}^{r\times m}$ is the user feature matrix, $\lambda\in[0,1]$ is the hyperparameter that measures the importance of each factorization term, and $\alpha>0$ is a balance hyperparameter.

After obtaining the optimized learned real-valued latent features, the recommendation is considered a similarity search problem. For instance, regarding a user's latent factor $\mathbf{u}_h$ as a query, we produce a ranking list of responses to the query. However, it is hard to make immediate response to effectively and efficiently match the items for the users.

To address this problem, we propose Binary Multi-modal Matrix Factorization (BMMF) method to replace the real-valued latent features with binary codes since the user-item similarity search can be efficiently conducted in the Hamming space. Denote $\mathbf{B}=[\mathbf{b}_1,\ldots,\mathbf{b}_m]\in\{\pm1\}^{r\times m},\mathbf{C}=[\mathbf{c}_1,\ldots,\mathbf{c}_{Lp}]\in\{\pm1\}^{r\times Lp}$, and $\mathbf{D}=[\mathbf{d}_1,\ldots,\mathbf{d}_n]\in\{\pm1\}^{r\times n}$ are user binary codes, mapping matrix binary codes of L modalities and item binary codes, respectively. To ensure that each bit splits the dataset as balanced as possible, a balanced constraint is imposed on the binary codes [34]. To make the binary codes more compact and each bit more independent, a decorrelated constraint is imposed on the binary codes. These two constraints ensure the information encoded in the short binary code [7]. Thus, we define our objective function as:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{C},\mathbf{D},\boldsymbol{\mu}}\ & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij}-\mathbf{d}_i^{\mathrm{T}}\mathbf{c}_j\right)^2+(1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih}-\mathbf{d}_i^{\mathrm{T}}\mathbf{b}_h\right)^2,\\ & \text{s.t.}\,\boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L=1,\boldsymbol{\mu}\geqslant\mathbf{0},\\ & \underbrace{\mathbf{B}\in\{\pm1\}^{r\times m},\mathbf{C}\in\{\pm1\}^{r\times Lp},\mathbf{D}\in\{\pm1\}^{r\times n}}_{\text{BinaryConstraint}},\\ & \underbrace{\mathbf{B1}=\mathbf{0},\mathbf{C1}=\mathbf{0},\mathbf{D1}=\mathbf{0}}_{\text{BalancedPartition}},\underbrace{\mathbf{BB}^{\mathrm{T}}=m\mathbf{I},\mathbf{CC}^{\mathrm{T}}=L_p\mathbf{I},\mathbf{DD}^{\mathrm{T}}=n\mathbf{I}}_{\text{Decorrelation}}.\end{aligned}\tag{6}$$

Since optimizing the above objective function is NP-hard, we propose to optimize the BMMF method by softening the balance and decorrelated constraints. To achieve this, we add three intermediate continuous variables: $\mathcal{B}=\left\{\mathbf{X}\in\mathbb{R}^{r\times m}|\mathbf{X1}=0,\mathbf{XX}^{\mathrm{T}}=m\mathbf{I}\right\}$, $\mathcal{D}=\left\{\mathbf{Y}\in\mathbb{R}^{r\times n}|\mathbf{Y1}=0,\mathbf{YY}^{\mathrm{T}}=n\mathbf{I}\right\},\mathcal{C}=\left\{\mathbf{Z}\in\mathbb{R}^{r\times Lp}|\mathbf{Z1}=0,\mathbf{ZZ}^{\mathrm{T}}=\ Lp\mathbf{I}\right\}$ and distances $d(\mathbf{B},\mathcal{B})=\min_{\mathbf{X}\in\mathcal{B}}\|\mathbf{B}-\mathbf{X}\|_F,d(\mathbf{D},\mathcal{D})=\min_{\mathbf{Y}\in\mathcal{D}}\|\mathbf{D}-\mathbf{Y}\|_F$, $d(\mathbf{C},\mathcal{C})=\min_{\mathbf{Z}\in\mathcal{C}}\|\mathbf{C}-\mathbf{Z}\|_F$. Therefore, we can transform the original BMMF in Eq. (6) as:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\mu}}\ & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij}-\mathbf{d}_i^{\mathrm{T}}\mathbf{c}_j\right)^2+(1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih}-\mathbf{d}_i^{\mathrm{T}}\mathbf{b}_h\right)^2\\ & +\alpha\left(\|\mathbf{B}-\mathbf{X}\|_F^2+\|\mathbf{C}-\mathbf{Z}\|_F^2+\|\mathbf{D}-\mathbf{Y}\|_F^2\right),\\ & \text{s.t.}\,\boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L=1,\boldsymbol{\mu}\geqslant\mathbf{0},\\ & \mathbf{B}\in\{\pm1\}^{r\times m},\mathbf{C}\in\{\pm1\}^{r\times Lp},\mathbf{D}\in\{\pm1\}^{r\times n},\\ & \mathbf{X1}=\mathbf{0},\mathbf{Z1}=\mathbf{0},\mathbf{Y1}=\mathbf{0},\mathbf{XX}^{\mathrm{T}}=m\mathbf{I},\mathbf{ZZ}^{\mathrm{T}}=Lp\mathbf{I},\mathbf{YY}^{\mathrm{T}}=n\mathbf{I},\end{aligned}\tag{7}$$

where $\alpha>0$ is a tuning parameter that indicates the degree of proximity between the binary codes and the intermediate real-valued representations. Since $\mathrm{tr}\left(\mathbf{BB}^{\mathrm{T}}\right)=\mathrm{tr}\left(\mathbf{XX}^{\mathrm{T}}\right)=mr,\mathrm{tr}\left(\mathbf{CC}^{\mathrm{T}}\right)=\mathrm{tr}\left(\mathbf{ZZ}^{\mathrm{T}}\right)=Lpr$, and$\mathrm{tr}\left(\mathbf{DD}^{\mathrm{T}}\right)=\mathrm{tr}\left(\mathbf{YY}^{\mathrm{T}}\right)=nr$ are all constants, Eq. (7) is equivalent to:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\mu}}\ & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij}-\mathbf{d}_i^{\mathrm{T}}\mathbf{c}_j\right)^2+(1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih}-\mathbf{d}_i^{\mathrm{T}}\mathbf{b}_h\right)^2\\ & -2\alpha\left(\mathrm{tr}\left(\mathbf{B}^{\mathrm{T}}\mathbf{X}\right)+\mathrm{tr}\left(\mathbf{C}^{\mathrm{T}}\mathbf{Z}\right)+\mathrm{tr}\left(\mathbf{D}^{\mathrm{T}}\mathbf{Y}\right)\right),\\ & \text{s.t.}\ \boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L=1,\boldsymbol{\mu}\geqslant\mathbf{0},\\ & \mathbf{B}\in\{\pm1\}^{r\times m},\mathbf{C}\in\{\pm1\}^{r\times Lp},\mathbf{D}\in\{\pm1\}^{r\times n},\\ & \mathbf{X1}=\mathbf{0},\mathbf{Z1}=\mathbf{0},\mathbf{Y1}=\mathbf{0},\mathbf{XX}^{\mathrm{T}}=m\mathbf{I},\mathbf{ZZ}^{\mathrm{T}}=Lp\mathbf{I},\mathbf{YY}^{\mathrm{T}}=n\mathbf{I}.\end{aligned}\tag{8}$$

It is worth noting that we do not discard the binary constraints in this objective transformation process. By jointly optimizing the binary codes and delegate real variables, we obtain the nearly balanced and decorrelated hash codes.

3.4. Item similarity preserving

The similarity between the items should be maintained when learning hash codes. This indicates that similar items in the original space should obtain similar binary codes with short Hamming distances.

Considering the above issues, we adopt the graph model [35] to efficiently preserve the similarities of the items during the binary embedding process. We define an affinity matrix $\mathbf{S} \in \{0,1\}^{n\times n}$ to represent the relationship of the items. Specifically, the similarity S_{ij} between item i and item j is defined as:

$$S_{ij} = \begin{cases} 1, & \text{if } \varphi(\mathbf{h})_i \in \mathcal{N}_k\left(\varphi(\mathbf{h})_j\right) \text{or} \varphi(\mathbf{h})_j \in \mathcal{N}_k(\varphi(\mathbf{h})_i), \\ 0, & \text{else}, \end{cases} \tag{9}$$

where $i,j = 1,2,\ldots,n, \mathcal{N}_k(\varphi(\mathbf{h}))$ is the set of k-nearest neighbours of $\varphi(\mathbf{h})$. To preserve the item similarity, we propose a semantic structure regularization term and formulate it as:

$$\min_{\mathbf{D}} \frac{1}{2}\sum_{i,j=1}^{n} \left\|\mathbf{d}_i - \mathbf{d}_j\right\|^2 S_{ij}. \tag{10}$$

3.5. Overall objective formulation

By integrating the above three parts into a unified learning framework, we obtain the overall objective formulation of the Binary Multi-modal Matrix Factorization (BMMF) as:

$$\begin{aligned} \min_{\mathbf{B},\mathbf{C},\mathbf{D},\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\mu}} \quad & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij} - \mathbf{d}_i^{\mathrm{T}}\mathbf{c}_j\right)^2 + (1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih} - \mathbf{d}_i^{\mathrm{T}}\mathbf{b}_h\right)^2 \\ & + \tfrac{1}{2}\omega\sum_{i,j=1}^{n}\left\|\mathbf{d}_i - \mathbf{d}_j\right\|^2 \mathbf{S}_{ij} \\ & -2\alpha\left(\mathrm{tr}\left(\mathbf{B}^{\mathrm{T}}\mathbf{X}\right) + \mathrm{tr}\left(\mathbf{C}^{\mathrm{T}}\mathbf{Z}\right) + \mathrm{tr}\left(\mathbf{D}^{\mathrm{T}}\mathbf{Y}\right)\right), \\ & \text{s.t.} \quad \boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L = 1, \boldsymbol{\mu} \geqslant \mathbf{0}, \\ & \mathbf{B} \in \{\pm 1\}^{r\times m}, \mathbf{C} \in \{\pm 1\}^{r\times Lp}, \mathbf{D} \in \{\pm 1\}^{r\times n}, \\ & \mathbf{X1} = \mathbf{0}, \mathbf{Z1} = \mathbf{0}, \mathbf{Y1} = \mathbf{0}, \mathbf{XX}^{\mathrm{T}} = m\mathbf{I}, \mathbf{ZZ}^{\mathrm{T}} = Lp\mathbf{I}, \mathbf{YY}^{\mathrm{T}} = n\mathbf{I}, \end{aligned} \tag{11}$$

where ω is the balance parameter.

3.6. Discrete optimization

We use an alternate optimization strategy to solve the mixed integer optimization problem. Specifically, we iteratively solve the BMMF model in Eq. (11) by taking turns to update each of $\boldsymbol{\mu}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$. Next, we show how to deal with each of the subproblems.

$\boldsymbol{\mu}$ – **subproblem**. In this subproblem, we aim to optimize $\boldsymbol{\mu}$ with fixed $\mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$. The optimization subproblem for $\boldsymbol{\mu}$ is:

$$\min_{\boldsymbol{\mu}} \left\|\varphi(\mathbf{H}) - \mathbf{D}^{\mathrm{T}}\mathbf{C}\right\|_F^2, \quad \text{s.t.} \boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L = 1, \boldsymbol{\mu} \geqslant \mathbf{0}. \tag{12}$$

The problem can be rewritten as follows:

$$\min_{\boldsymbol{\mu}} \sum_{l=1}^{L}\mu_l\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2, \quad \text{s.t.} \sum_{l=1}^{L}\mu_l = 1, \quad \mu_l \geqslant 0, \tag{13}$$

where $\mu_l = 1$ corresponds to $\min\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2$ over all modalities, and $\mu_l = 0$ otherwise. In this situation, the optimization result of the above formula would select only one modality, while ignoring the complementary information of the other modalities. To overcome this limitation, we introduce a penalty exponent $t > 1$ and set $\mu_l \leftarrow \mu_l^t$. The new optimization problem is reformulated as:

$$\begin{aligned} & \min_{\boldsymbol{\mu}} \sum_{l=1}^{L}\mu_l^t\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2, \\ & \text{s.t.} \quad \sum_{l=1}^{L}\mu_l = 1, \mu_l \geqslant 0. \end{aligned} \tag{14}$$

As proven in [36], the Lagrange function of Eq. (14) can be constructed as:

$$\mathcal{L}(\boldsymbol{\mu}, \xi) = \sum_{l=1}^{L}\mu_l^t\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2 - \xi\left(\sum_{l=1}^{L}\mu_l - 1\right), \tag{15}$$

where ξ is a hyper-parameter that balances the regularization terms.

We can obtain μ_l as:

$$\mu_l = \frac{\left(1/\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2\right)^{1/(t-1)}}{\sum_{l=1}^{L}\left(1/\left\|\varphi\left(\mathbf{H}^l\right) - \mathbf{D}^{\mathrm{T}}\mathbf{C}^l\right\|_F^2\right)^{1/(t-1)}}. \tag{16}$$

$\mathbf{B}$ – **subproblem**. In this subproblem, we update $\mathbf{B}$ with fixed $\boldsymbol{\mu}, \mathbf{C}, \mathbf{D}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$. Since the objective function in Eq. (11) is obtained by summing over independent users, we can update $\mathbf{B}$ by updating $\mathbf{b}_h$ in parallel according to:

$$\begin{aligned} & \underset{\mathbf{b}_h\in\{\pm 1\}^r}{\operatorname{argmin}}(1-\lambda)\left(\mathbf{b}_h^{\mathrm{T}}\left(\sum_{j\in\Psi_h}\mathbf{d}_i\mathbf{d}_i^{\mathrm{T}}\right)\mathbf{b}_h - 2\left(\sum_{i\in\Psi_h}r_{ih}\mathbf{d}_i^{\mathrm{T}}\right)\mathbf{b}_h\right) \\ & - 2\alpha\mathbf{x}_h^{\mathrm{T}}\mathbf{b}_h, \end{aligned} \tag{17}$$

where Ψ_h is the items set rated by user h.

Considering the binary constraints, Discrete Coordinate Descent (DCD) is used to update the binary codes $\mathbf{b}_h$. In particular, let b_{hk} be the k-th bit of $\mathbf{b}_h$ and $\mathbf{b}_{h\bar{k}}$ be the rest codes excluding b_{hk}, DCD can update b_{hk} with $\mathbf{b}_{h\bar{k}}$ fixed. We update b_{hk} based on the following rule:

$$\begin{aligned} & b_{hk} \leftarrow \mathrm{sgn}\left(K\left(\hat{b}_{hk}, b_{hk}\right)\right), \\ & \hat{b}_{hk} = (1-\lambda)\sum_{i\in\Psi_h}\left(r_{ih} - \mathbf{d}_{i\bar{k}}^{\mathrm{T}}\mathbf{b}_{h\bar{k}}\right)d_{ik} + \alpha x_{hk}, \end{aligned} \tag{18}$$

where $\mathbf{d}_{i\bar{k}}^{\mathrm{T}}$ is the remaining set of item codes excluding d_{ik}, and $K(x,y)$ is a function where $K(x,y) = x$ if $x \neq 0$ and $K(x,y) = y$ otherwise. We do not update b_{hk} when $\hat{b}_{hk}$ is zero. The update rule is applied to all the hash bits iteratively until convergence.

$\mathbf{C}$ – **subproblem**. In this subproblem, we update $\mathbf{C}$ with fixed $\boldsymbol{\mu}, \mathbf{B}, \mathbf{D}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$. Similar to the B-subproblem, we can update $\mathbf{C}$ by updating $\mathbf{c}_j$ in parallel according to:

$$\underset{\mathbf{c}_j\in\{\pm 1\}^r}{\operatorname{argmin}}\lambda\left(\mathbf{c}_j^{\mathrm{T}}\left(\sum_{i\in\Omega_j}\mathbf{d}_i\mathbf{d}_i^{\mathrm{T}}\right)\mathbf{c}_j - 2\left(\sum_{i\in\Omega_j}\varphi(\mathbf{H})_{ij}\mathbf{d}_i^{\mathrm{T}}\right)\mathbf{c}_j\right) - 2\alpha\mathbf{z}_j^{\mathrm{T}}\mathbf{c}_j, \tag{19}$$

where Ω_j is the items set associated with multi-modal feature j.

Denoting c_{jk} as the k-th bit of $\mathbf{c}_j$ and $\mathbf{c}_{j\bar{k}}$ as the remaining codes excluding c_{jk}, we update c_{jk} based on the following rule:

$$\begin{aligned} & c_{jk} \leftarrow \mathrm{sgn}(K(\hat{c}_{jk}, c_{jk})), \\ & \hat{c}_{jk} = \lambda\sum_{i\in\Omega_j}\left(\varphi(\mathbf{H})_{ij} - \mathbf{d}_{i\bar{k}}^{\mathrm{T}}\mathbf{c}_{j\bar{k}}\right)d_{ik} + \alpha z_{jk}. \end{aligned} \tag{20}$$

$\mathbf{D}$ – **subproblem**. In this subproblem, we update $\mathbf{D}$ with fixed $\boldsymbol{\mu}, \mathbf{B}, \mathbf{C}, \mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$. Similar to the B-subproblem, we can update $\mathbf{D}$ by updating $\mathbf{d}_i$ in parallel according to:

$$\begin{aligned}\underset{\mathbf{d}_i\in\{\pm1\}^r}{\operatorname{argmin}} \quad & \lambda\left(\mathbf{d}_i^{\mathrm{T}}\left(\sum_{j\in\Omega_i}\mathbf{c}_j\mathbf{c}_j^{\mathrm{T}}\right)\mathbf{d}_i - 2\left(\sum_{j\in\Omega_i}\varphi(\mathbf{H})_{ij}\mathbf{c}_j^{\mathrm{T}}\right)\mathbf{d}_i\right) \\ & +(1-\lambda)\left(\mathbf{d}_i^{\mathrm{T}}\left(\sum_{h\in\Psi_i}\mathbf{b}_h\mathbf{b}_h^{\mathrm{T}}\right)\mathbf{d}_i - 2\left(\sum_{h\in\Psi_i}r_{ih}\mathbf{b}_h^{\mathrm{T}}\right)\mathbf{d}_i\right) \\ & -\omega\sum_{i,j=1}^{n}S_{ij}\mathbf{b}_i^{\mathrm{T}}\mathbf{d}_j - 2\alpha\mathbf{y}_i^{\mathrm{T}}\mathbf{d}_i,\end{aligned} \tag{21}$$

Denoting d_{ik} as the k-th bit of $\mathbf{d}_i$ and $\mathbf{d}_{i\bar{k}}$ as the remaining codes excluding d_{ik}, we update d_{ik} based on the following rule:

$$\begin{aligned} d_{ik} &\leftarrow \operatorname{sgn}\left(K\left(\hat{d}_{ik}, d_{ik}\right)\right), \\ \hat{d}_{ik} &= \lambda\sum_{j\in\Omega_i}\left(\varphi(\mathbf{H})_{ij} - \mathbf{c}_{j\bar{k}}^{\mathrm{T}}\mathbf{d}_{i\bar{k}}\right)c_{jk} \\ &\quad +(1-\lambda)\sum_{h\in\Psi_i}\left(r_{ih} - \mathbf{b}_{h\bar{k}}^{\mathrm{T}}\mathbf{d}_{h\bar{k}}\right)b_{hk} + \tfrac{\omega}{2}\sum_{i,j=1}^{n}S_{ij}d_{jk} + \alpha y_{ik}.\end{aligned} \tag{22}$$

$\mathbf{X}$ – **subproblem**. In this subproblem, we update $\mathbf{X}$ with fixed $\boldsymbol{\mu}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Y}$ and $\mathbf{Z}$, and the optimization subproblem for $\mathbf{X}$ is:

$$\underset{\mathbf{X}}{\operatorname{argmax}} \operatorname{tr}\left(\mathbf{B}^{\mathrm{T}}\mathbf{X}\right), \text{ s.t. } \mathbf{X1} = 0, \mathbf{XX}^{\mathrm{T}} = m\mathbf{I}. \tag{23}$$

The optimal solution of Eq. (23) can be addressed by a Singular Value Decomposition (SVD) according to [10]. Specifically, we can obtain a closed-form solution of $\mathbf{X}$ as:

$$\mathbf{X} \leftarrow \sqrt{m}\left[\mathbf{P}_b\ \widehat{\mathbf{P}}_b\right]\left[\mathbf{Q}_b\ \widehat{\mathbf{Q}}_b\right]^{\mathrm{T}}, \tag{24}$$

where $\mathbf{P}_b$ and $\mathbf{Q}_b$ are the left and right singular vectors, respectively, which are obtained by an SVD of the row-centered matrix $\overline{\mathbf{B}} : \bar{b}_{ij} = b_{ij} - \frac{1}{m}\sum_{i=1}^{m} b_{ij}$, $\widehat{\mathbf{P}}_b$ is stacked by the left singular vectors. In addition, we calculate $\widehat{\mathbf{Q}}_b$ by Gram-Schmidt orthogonalization based on $[\mathbf{Q}_b\ \mathbf{1}]$.

$\mathbf{Y}$ – **subproblem**. In this subproblem, we update $\mathbf{Y}$ with fixed $\boldsymbol{\mu}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{X}$ and $\mathbf{Z}$. The optimization subproblem for $\mathbf{Y}$ is:

$$\underset{\mathbf{Y}}{\operatorname{argmax}} \operatorname{tr}\left(\mathbf{D}^{\mathrm{T}}\mathbf{Y}\right), \text{ s.t. } \mathbf{Y1} = 0, \mathbf{YY}^{\mathrm{T}} = n\mathbf{I}. \tag{25}$$

Similar to the X-subproblem, we can obtain a closed-form solution as follows:

$$\mathbf{Y} \leftarrow \sqrt{n}\left[\mathbf{P}_d\ \widehat{\mathbf{P}}_d\right]\left[\mathbf{Q}_d\ \widehat{\mathbf{Q}}_d\right]^{\mathrm{T}}. \tag{26}$$

$\mathbf{Z}$ – **subproblem**. In this subproblem, we update $\mathbf{Z}$ with fixed $\boldsymbol{\mu}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{X}$ and $\mathbf{Y}$. The optimization subproblem for $\mathbf{Z}$ is:

$$\underset{\mathbf{Z}}{\operatorname{argmax}} \operatorname{tr}\left(\mathbf{C}^{\mathrm{T}}\mathbf{Z}\right), \text{ s.t. } \mathbf{Z1} = 0, \mathbf{ZZ}^{\mathrm{T}} = Lp\mathbf{I}. \tag{27}$$

Similar to the X-subproblem, we can obtain a closed-form solution as follows:

$$\mathbf{Z} \leftarrow \sqrt{Lp}\left[\mathbf{P}_c\ \widehat{\mathbf{P}}_c\right]\left[\mathbf{Q}_c\ \widehat{\mathbf{Q}}_c\right]^{\mathrm{T}}. \tag{28}$$

We iteratively repeat the above optimization procedures until convergence.

3.7. Better initialization for the discrete optimization problems

Since our BMMF model involves a discrete matrix factorization problem, a better initialization algorithm usually contributes to achieving faster convergence and ideal local optimal solutions. Inspired by DCF [10], we suggest an effective initialization strategy, which solves the relaxed problem of Eq. (11) as follows:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\mu}} \quad & \lambda\sum_{i,j\in\Omega}\left(\varphi(\mathbf{H})_{ij} - \mathbf{v}_i^{\mathrm{T}}\mathbf{w}_j\right)^2 + (1-\lambda)\sum_{i,h\in\Psi}\left(r_{ih} - \mathbf{v}_i^{\mathrm{T}}\mathbf{u}_h\right)^2 \\ & +\tfrac{1}{2}\omega\sum_{i,j=1}^{n}\left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2\mathbf{S}_{ij} \\ & -2\alpha\left(\operatorname{tr}\left(\mathbf{U}^{\mathrm{T}}\mathbf{X}\right) + \operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{Z}\right) + \operatorname{tr}\left(\mathbf{V}^{\mathrm{T}}\mathbf{Y}\right)\right) \\ & +\alpha\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{W}\|_F^2\right), \\ & \text{s.t.} \boldsymbol{\mu}^{\mathrm{T}}\mathbf{1}_L = 1,\ \boldsymbol{\mu} \geqslant \mathbf{0}, \\ & \mathbf{X1} = \mathbf{0}, \mathbf{Z1} = \mathbf{0}, \mathbf{Y1} = \mathbf{0}, \mathbf{XX}^{\mathrm{T}} = m\mathbf{I}, \mathbf{ZZ}^{\mathrm{T}} = Lp\mathbf{I}, \mathbf{YY}^{\mathrm{T}} = n\mathbf{I}.\end{aligned} \tag{29}$$

We solve the above initialization with balanced and decorrelated constraints imposed on real-valued $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$, leading to a small quantization error. To solve Eq. (29), we use alternating optimization to solve the $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ subproblems, and take the SVD method to solve the $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ subproblems.

Algorithm 1 Learning algorithm for BMMF

Input: Observed user-item ratings $\mathbf{R} \in \mathbb{R}^{n\times m}$, multi-modal features matrix of items $\left\{\mathbf{H}^l \in \mathbb{R}^{n\times d_l}\right\}_{l=1}^{L}$, the number of anchors p, the Gaussian kernel band-width σ_l, the hash code length r, parameters λ, α, ω.
Output: Binary user hash codes $\mathbf{B} \in \mathbb{R}^{r\times m}$, binary item codes $\mathbf{D} \in \mathbb{R}^{r\times n}$.

```
Initialize μ_l = 1/L.
Randomly select the anchors from each modality.
Construct the consensus multi-modal feature φ(H) as Eq. (3).
B, C, D, X, Y, Z are initialized according to the initialization strategy proposed in Section 3.7.
while not convergence do
  for h ∈ {1, ..., m} do
    repeat
      Update b_h bit by bit according to Eq. (17) (r bits in total).
    until convergence
  end for
  for j ∈ {1, ..., L_p} do
    repeat
      Update c_j bit by bit according to Eq. (19) (r bits in total).
    until convergence
  end for
  for i ∈ {1, ..., n} do
    repeat
      Update d_i bit by bit according to Eq. (21) (r bits in total).
    until convergence
  end for
  Update μ according to Eq. (16).
  Update X according to Eq. (25).
  Update Y according to Eq. (27).
  Update Z according to Eq. (29).
end while
```

Assuming the solutions are denoted by $\mathbf{U}^*, \mathbf{V}^*, \mathbf{W}^*, \mathbf{X}^*, \mathbf{Y}^*, \mathbf{Z}^*$, we can obtain the initial binary code as follows:

$$\begin{aligned}&\mathbf{B} \leftarrow \operatorname{sgn}(\mathbf{U}^*), \mathbf{D} \leftarrow \operatorname{sgn}(\mathbf{V}^*), \mathbf{C} \leftarrow \operatorname{sgn}(\mathbf{W}^*), \\ &\mathbf{X} \leftarrow \mathbf{X}^*, \mathbf{Y} \leftarrow \mathbf{Y}^*, \mathbf{Z} \leftarrow \mathbf{Z}^*.\end{aligned} \tag{30}$$

Table 2
Statistics of experimental datasets.

Datasets	#User	#Item	#Rating	Sparsity
BookCrossing	2,151	6,830	180,595	98.77%
MovieLens-1M	6,040	3,706	1,000,209	95.53%
MovieLens-10M	71,567	10,681	10,000,054	98.66%

The above initialization is obviously feasible for Eq. (11). The convergence of the objective functions with the above initialization is demonstrated on three datasets as shown in Fig. 7. The pseudo-code for model updating is given in Algorithm 1.

3.8. Computing complexity analysis

The time complexity of constructing the consensus multi-modal feature $\varphi(\mathbf{H})$ is $\mathcal{O}(nLp)$. It takes $\mathcal{O}(L)$ for updating $\boldsymbol{\mu}$. We assume that T_1 is the number of outer loop iterations needed for convergence (Step 7 to 9), T_2 is the number of inner loop of updating $\mathbf{b}_h, \mathbf{c}_j$ and $\mathbf{d}_i$ are needed for convergence, then the overall complexity for the $\mathbf{B}$-subproblem is $\mathcal{O}(r^2T_2|\Psi|)$. Similarly, the overall complexity for the $\mathbf{C}$-subproblem and $\mathbf{D}$-subproblem are $\mathcal{O}(r^2T_2|\Omega|)$ and $\mathcal{O}(r^2T_2(|\Omega|+|\Psi|))$, respectively. The $\mathbf{X}$-subproblem requires $\mathcal{O}(r^2m)$ to update. Similarly, it takes $\mathcal{O}(r^2n)$ and $\mathcal{O}(r^2L_p)$ to solve the $\mathbf{Y}$-subproblem and $\mathbf{Z}$-subproblem, respectively. Hence, the overall time complexity for Algorithm 1 is $\mathcal{O}(T_1(T_2(r^2(|\Omega| +|\Psi|))) + r^2(m+n+L_p)))$. In summary, training the BMMF method is linear with the size of the dataset, which is scalable for large-scale recommendation.

4. Experiments

4.1. Evaluation datasets

The experiments are conducted on three publicly available datasets: BookCrossing,[4] MovieLens-1M,[5] and MovieLens-10M[5]. In these three datasets, each user has only one rating for each item.

- **BookCrossing** includes 1,149,780 ratings (containing implicit and explicit feedback) from 278,858 users for 271,379 books. The rating is expressed on a scale from 1 to 10 with 1 granularity for explicit feedback and 0 for implicit feedback. Most books in this dataset have content-based information including the title, author and publication.
- **MovieLens-1 M** consists of 1,000,209 ratings from 6,040 users for 3,706 movies collected from the MovieLens website by GroupLens Research. All its rating scores are integers from 1 to 5 with 1 as the interval. All movies in this dataset are associated with the titles and genre labels.
- **MovieLens-10 M** is comprised of 71,567 users, 10,681 movies and 10,000,054 ratings. The scale of the rating scores ranged from 0.5 to 5 with a step of 0.5. In addition, most movies are related to 3–5 tags from a dictionary of 18 genre tags in this dataset.

Similar to [11], considering the serious sparsity of the original BookCrossing dataset, we remove the users and items with less than 20 ratings. After filtering, the BookCrossing dataset contains 2,151 users, 6,830 items, and 180,595 ratings. For the MovieLens-1M and MovieLens-10M datasets, we keep all the users and items without any filtering. Table 2 sums up the statistics of these datasets. Similar to [32], we obtain the feature representations of the users by the one-hot encoding approach and extract the auxiliary information of the items by the bag-of-words encoding method. In the experiments, we design two types of recommendation settings: 1) Cold-start scenario: we randomly select 20% of each user's ratings as the training set and the rest as the testing set. 2) General scenario: we split each user's items into training and testing sets using a 50/50 split. We independently repeat the experiments five times and the average results are reported.

4.2. Evaluation metrics

Similar to the existing methods [10,11,14], we adopt the Normalized Discounted Cumulative Gain (NDCG) to evaluate the recommendation performance, which takes into account both the ranking precisions and rating positions for evaluating recommendation performances [37]. Generally, higher NDCG values indicate a better ranking accuracy.

4.3. Evaluation baselines

In this paper, we compare our approach with a state-of-the-art continuous value-based recommendation method and four hashing-based recommendation methods.

- **Discrete Content-aware Matrix Factorization (DCMF)** [11] is an advanced binarized method for collaborative filtering with content information. It is an extension of DCF that encodes the auxiliary information of each user and item into binary codes. The parameters λ_1 and λ_2 for modelling the user and item textual features are tuned within the range of $\{1, 10, 50, 100, 500, 1000\}$.
- **Discrete Factorization Machines (DFM)** [14] is the first binarized factorization machines method. It is capable of modelling the pair-wise interaction between the features and learning the binary embedding for each item. The parameter β for the softened decorrelation constraint is tuned within the range of $\left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\right\}$.
- **Discrete Deep Learning (DDL)** [13] is a binary deep recommendation approach. It combines the Deep Belief Network (DBN) with collaborative filtering framework to learn binary codes based on the side information. The parameters α and β are tuned within the range of $\left\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\right\}$, and λ within the range of $\left\{10^{-1}, 1, 10, 100\right\}$. The layer structure of the DBN is set in the range of $\{8000, 800, 30\}$.
- **Multi-feature Discrete Collaborative Filtering (MFDCF)** [32] pr-ojects the multiple content features of the users into the binary hash codes to support fast cold-start recommendation. The parameters α, β and γ are tuned within the range of $\left\{10^{-6}, 10^{-4}, 10^{-1}, 1, 10, 10^3, 10^6\right\}$.
- **Low-rank Linear Auto-Encoder (LLAE)** [38] first applies zero-shot learning [39] to the cold-start recommendation problem. It extracts the user preferences for each item from the user attributes. The parameters λ and β for the relax and low-rank constraints are tuned within the range of $\left\{10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 100\right\}$.

4.4. Implementation details

The proposed BMMF has several parameters: λ, α, ω, and the number of anchors p. λ is a tuning parameter that measures the contribution of each factorization. α and ω control the balance between the regularization terms when learning hash codes. The

[4] https://grouplens.org/datasets/book-crossing/.
[5] https://grouplens.org/datasets/movielens/.

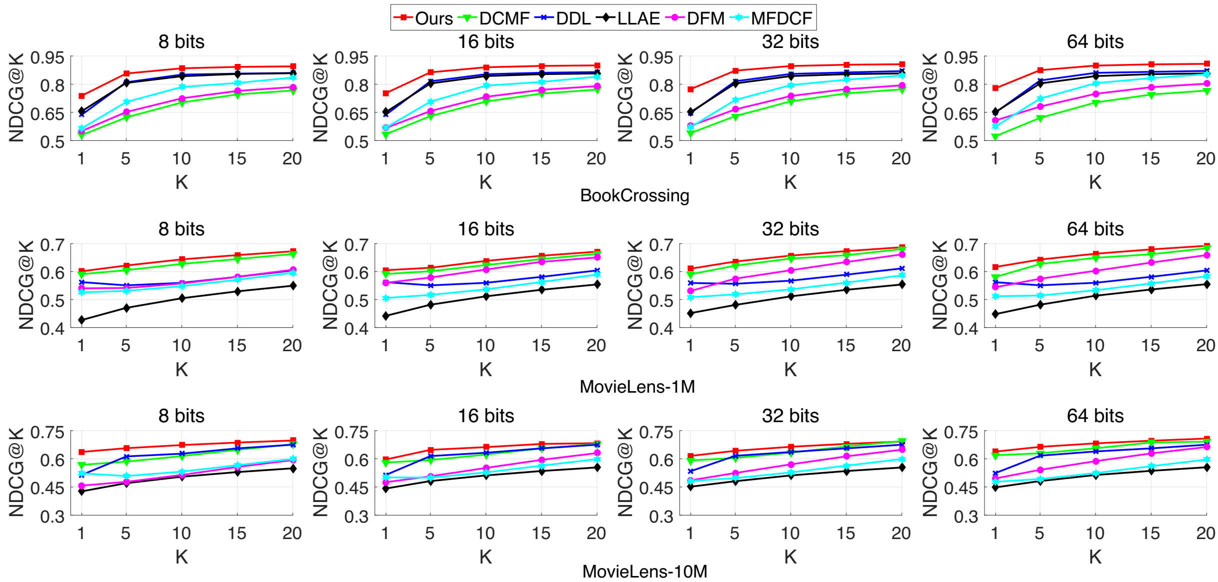

Fig. 2. The NDCG@K performance of all methods on BookCrossing, MovieLens-1M, and MovieLens-10M in the cold-start scenario.

optimal hyper parameters of the BMMF method are tuned by grid search. For λ, we tune it in the range of $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For α, we tune it in the range of $\left\{10^{-6}, 10^{-3}, 1, 10^{3}, 10^{6}\right\}$ while fixing the other parameters. For ω, we tune it in the range of $\left\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\right\}$. For p, we tune it from 100 to 1000 with an interval of 100. The best performance of the proposed BMMF is achieved when $\{\lambda = 0.9, \alpha = 1, \omega = 0.001, p = 400\}$, $\{\lambda = 0.9, \alpha = 0.01, \omega = 0.01, p = 300\}$, $\{\lambda = 0.9, \alpha = 0.1, \omega = 0.01, p = 300\}$ on BookCrossing, MovieLens-1M, and MovieLens-10M, respectively. For the baselines, we carefully tune their parameters and finally report their best results for the performance comparison.

4.5. Accuracy comparison

This experiment studies the recommendation performance between the baselines and BMMF in an item cold-start recommendation setting. Our focus is to test whether the marked-off cold-start items can be accurately recommended to the appropriate user. Fig. 2 shows the recommendation performance (NDCG@1 to NDCG@20) of BMMF and the baseline methods on the three datasets. The hash code length ranges from 8 to 64 bits. From Fig. 2, we observe that BMMF outperforms the other baselines. The reasons are as follows: DCMF learns the binary codes for each user and item from the auxiliary information. Nevertheless, it does not take into account the similarities between the items. DFM is a generic

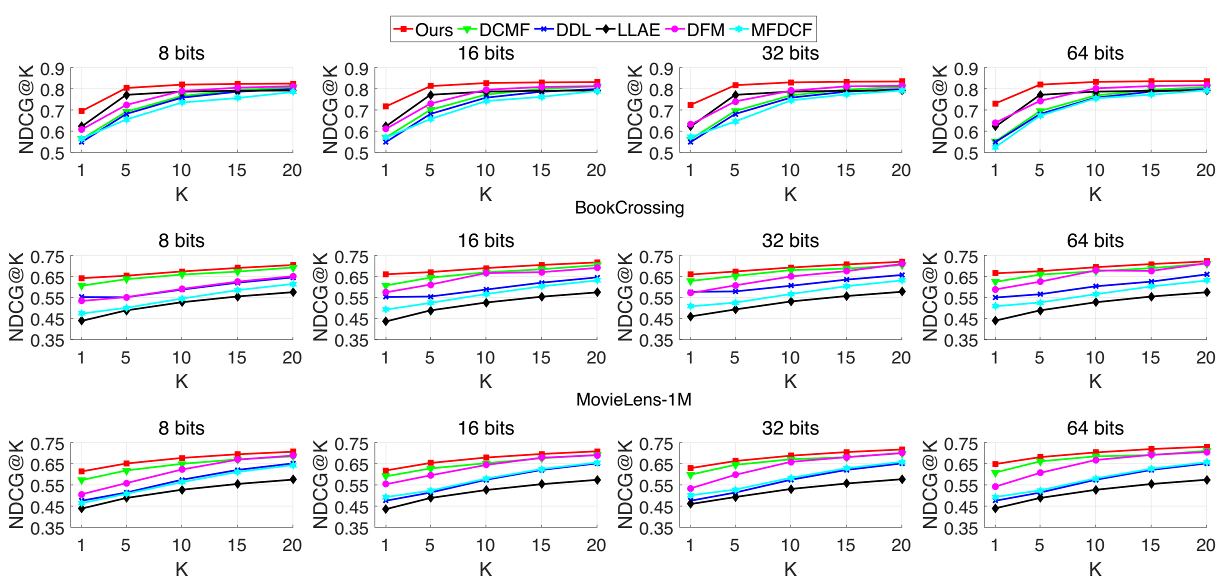

Fig. 3. The NDCG@K performance of all methods on BookCrossing, MovieLens-1M, and MovieLens-10M in the general scenario(non-cold-start).

Table 3
Efficiency comparison between BMMF and other hashing-based recommendation methods when the code length ranges from 8 to 64 on the MovieLens-1M.

Methods	Initialization time/iteration (s)				Training time/iteration (s)			
	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
DDL	6.2311	6.3524	6.5689	6.8724	30.9573	66.9637	146.9787	381.7193
DFM	18.5609	19.1072	21.6826	27.3214	14.6326	19.2502	21.0988	25.1534
DCMF	1.3013	1.3842	1.6467	2.4625	1.2679	1.3671	1.6062	2.1156
MFDCF	0.0615	0.0868	0.1561	0.3030	0.0590	0.0870	0.1576	0.3053
Ours	0.9751	1.3484	2.9866	6.8473	1.4625	1.6839	2.9098	9.7562

feature-based recommendation model that learns a binary embedding for each feature. However, it ignores collaborative interactions. DDL combines DBN and collaborative filtering framework to learn the binary codes for the users and items. However, the learning process of the DBN is independent of the overall optimization process, which leads to sub-optimal results. MFDCF is a unified multi-feature discrete collaborative filtering method that adaptively projects the multiple content features into binary hash codes. However, it does not take the similarity preservation of the items into account, and its discrete optimization strategy also results in a larger quantization loss. Moreover, the results show that our proposed BMMF is consistently better than the continuous value based hybrid recommendation method LLAE due to our consideration of consensus multi-modal feature mapping.

Additionally, we compare the performance of BMMF and baseline methods in a general setting, and the experimental results are shown in Fig. 3, from which, we observe that the BMMF outperforms the other baselines, demonstrating the recommendation performance of BMMF in a general setting.

4.6. Run time comparison

In the experiments, we compare the computational efficiency of our proposed BMMF with four state-of-the-art hashing-based recommendation methods and report the experimental results on MovieLens-1M in Table 3. Similar results can be obtained on the other datasets. The compared approaches are all implemented with MATLAB. Table 3 demonstrates the efficiency of these methods in both the initialization stage and training stage using a 3.40 GHz Intel(R) Core(TM) i7-6700 CPU. From Table 3, we observe that MFDCF achieves significant speedups. It develops an augmented Lagrangian multiplier to directly calculate hash codes by simple and efficient operations. However, from Fig. 2, we also find that this optimization method will result in more information loss than the bit-by-bit optimization strategy. DDL needs to pre-train the DBN in the initialization stage, and updates the DBN to generate hash codes of the items in each round of iteration at the training stage, slowing down the model training considerably. Compared with DDL, BMMF is approximately 1 to 6 times faster than DDL in the initialization stage and approximately 15 to 39 times faster in the training stage. The main cost of DFM is modelling the pairwise interactions between any content features, which makes it highly intractable in large-scale settings.

Compared with DFM, our proposed BMMF is approximately 4 to 19 times faster than DFM in the initialization stage and approximately 2 to 11 times faster in the training stage. Compared with DCMF, our proposed BMMF is faster than DCMF in the initialization stage when the code length ranges from 8 to 16. With longer hash codes, the training process of BMMF is slower than that of DCMF. The main reason for this problem is that BMMF obtains the hash codes from the multi-modal auxiliary features and interaction features of the users and items, and large amounts of training data reduces the efficiency of BMMF. The experimental results in Fig. 2 and Table 3 demonstrate the superiority of BMMF over the other state-of-the-art methods in recommendation efficiency and cold-start recommendation accuracy.

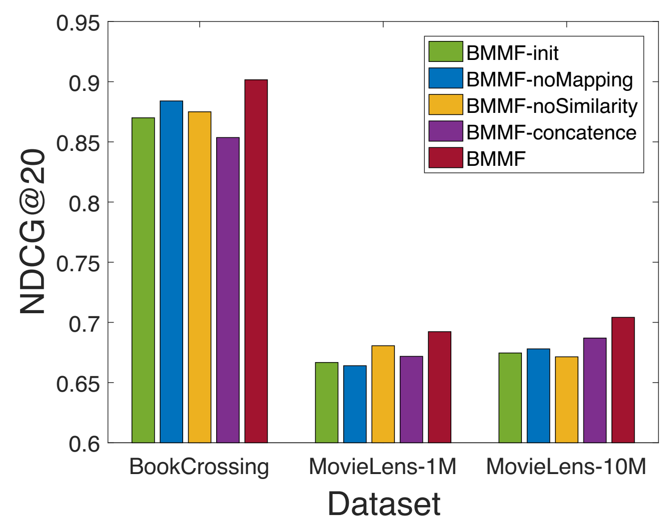

Fig. 4. NDCG@20 obtained by the variants of the proposed method BMMF (code length $r = 16$) in item cold-start recommendation scenarios. BMMF-init randomly initializes the relevant variables without the training process. BMMF-nomapping directly inputs the original features of different modalities into the model to validate the effects of nonlinear mapping. BMMF-noSimilarity simply removes the regularization term of similarity preservation. BMMF-concatenate simply concatenates multiple features from different modalities as a unified vector and inputs it into the model.

4.7. Ablation analysis

4.7.1. Effects of the proposed initialization strategy

In BMMF, we propose an effective initialization strategy, which is important for fast convergence. To validate its effect, a variant BMMF-init is designed for comparison, in which the relevant variables are randomly initialized without the training process. Fig. 4 shows a comparison of the item cold-start recommendation performance. It is easily observed that the BMMF consistently performs better than BMMF-init on the three datasets, showing the effectiveness of the initialization strategy in BMMF. Fig. 7 plots the objective function value with the number of iterations on three datasets. The experimental results demonstrate that BMMF with initialization can converge rapidly.

4.7.2. Effects of concensus multi-modal feature mapping

In BMMF, we project the features of different modalities into unified representations by anchor-based nonlinear mapping. This process can effectively capture the nonlinear structure of the original features. To verify its effect, a variant BMMF-noMapping is designed for comparison, which takes the original features of the different modalities as input. The comparison of the item cold-start recommendation performance is plotted in Fig. 4. We observe that the accuracy of our method markedly outperforms BMMF-

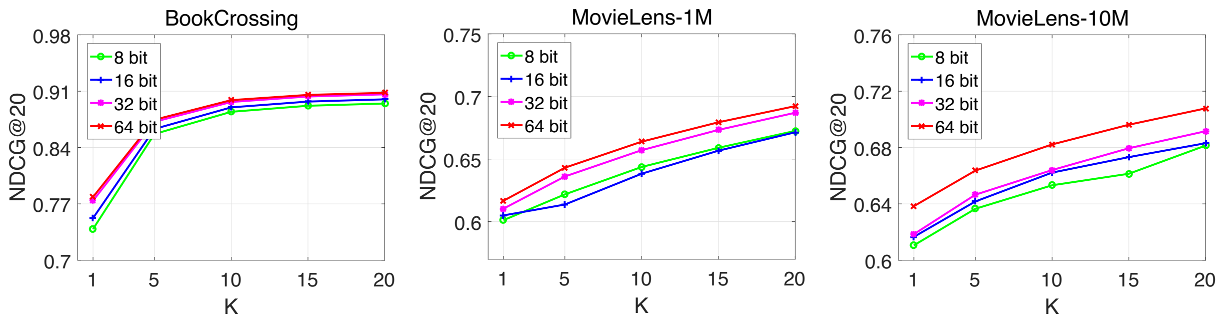

Fig. 5. Performance variations with hash code length on three datasets in cold-start setting.

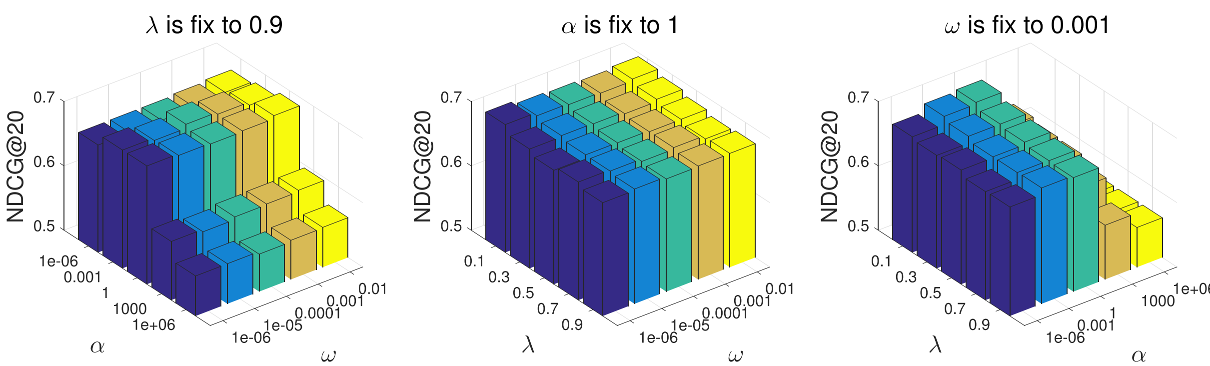

Fig. 6. Performance variations with the key parameters on MovieLens-1M.

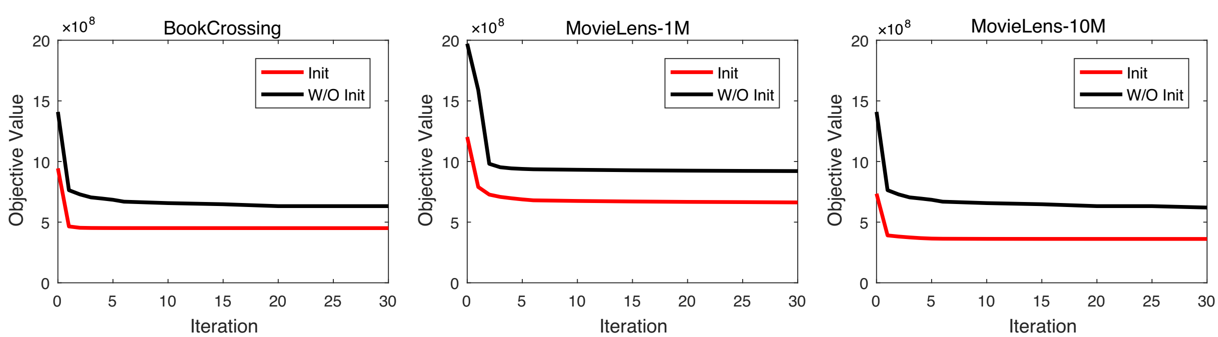

Fig. 7. The variations (with/without initialization) of objective function value on three datasets when the hash code length is 16 bits.

noMapping. The experimental results demonstrate that the consensus multi-modal feature mapping in BMMF can indeed improve the recommendation accuracy.

4.7.3. Effects of mutli-modal feature fusion

In BMMF, we propose a multi-modal fusion strategy to preserve the multi-modal auxiliary information into hash codes while exploiting their complementarity. To validate its effectiveness, we evaluate a variant of BMMF called BMMF-concatenate for comparison. It simply concatenates multiple features from different modalities as a unified vector and then imports it into the hashing model. The accuracy comparison is plotted in Fig. 4. From it, we observe that BMMF can achieve higher accuracy than the BMMF-concatenate. The experimental results demonstrate that the multi-modal fusion strategy can effectively explore the complementarity of the multi-modal auxiliary information and improve the recommendation accuracy.

4.7.4. Effects of item similarity preservation

We design a semantic structure regularization term in BMMF to enhance the item similarity preservation. To validate the effect of the similarity preservation part, a variant BMMF-noSimilarity is designed for comparison, which removes the regularization term of the similarity preservation. The recommendation performance of BMMF-noSimilarity and BMMF are compared in Fig. 4, where we can observe that the precision of BMMF is clearly higher than that of BMMF-noSimilarity on three datasets. The experimental results demonstrate that our proposed semantic structure regularization term performs well on preserving the similarity of the items and improving the recommendation performance.

4.8. Effects of hash code length

In this section, we conduct experiments to observe the performance variations with hash code lengths. Fig. 5 shows the NDCG@20 on different datasets with varying hash code lengths in the cold-start setting. From the results, we find that the performance of BMMF improves as the hash code length increases. This is mainly because BMMF can maximize the information encoded in finite-length hash codes, which leads to a stable performance improvement.

4.9. Parameter sensitivity analysis and convergency experiments

We conduct experiments to observe the performance variations with the involved parameters λ, α, and ω. Fig. 6 plots the detailed experimental results on MovieLens-1M when the hash code length is 16 bits. Similar experimental results can be obtained on other datasets and hash code lengths. From Fig. 6, we observe that BMMF achieves satisfactory performance over a wide range of λ and ω. We also find that BMMF performs superiorly and steadily when α is in the range of $\{10^{-6}, \ldots, 1\}$. To validate the convergency of BMMF, we perform further experimental analysis on the three datasets when the hash code length is 16 bits. The convergence curves recording the objective function of BMMF are plotted in Fig. 7. The experimental results demonstrate that BMMF can converge rapidly.

5. Conclusion and future work

In this paper, we propose a Binary Multi-modal Matrix Factorization (BMMF) method for efficient item cold-start recommendation. We develop a unified binary multi-modal matrix factorization framework that takes into account the item-item similarities, user-item interactions, and multi-modal features of the items to learn the binary hash codes. In addition, we develop an efficient discrete optimization method to iteratively solve the formulated discrete hash optimization problem. Experimental results on three public datasets demonstrate that our approach can achieve superior recommendation performance. As the data is generated at an unprecedented rate every day, the recommender system should be able to capture the real-time interests of the users timely, so we plan to extend the proposed BMMF to an online matrix factorization model in the future.

CRediT authorship contribution statement

Chengmei Peng: Conceptualization, Methodology, Writing – original draft. **Lei Zhu:** Conceptualization, Methodology, Writing – review & editing. **Yang Xu:** Conceptualization, Methodology, Writing – review & editing. **Yaping Li:** Validation, Software. **Lei Guo:** Visualization, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62172263 and Grant 62002209, in part by the Natural Science Foundation of Shandong, China, under Grant ZR2020YQ47, Grant ZR2020QF111 and Grant ZR2019QF002, in part by the Youth Innovation Project of Shandong Universities, China, under Grant 2019KJN040.

References

[1] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) 1–38.
[2] Y. Li, D. Wang, H. He, L. Jiao, Y. Xue, Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems, Neurocomputing 249 (2017) 48–63.
[3] Z. Yang, W. Chen, J. Huang, Enhancing recommendation on extremely sparse data with blocks-coupled non-negative matrix factorization, Neurocomputing 278 (2018) 126–133.
[4] X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank: Review-aware explainable recommendation by modeling aspects, in: Proceedings of the ACM International on Conference on Information and Knowledge Management (CIKM), 2015, pp. 1661–1670.
[5] J. Wang, T. Zhang, J. Song, N. Sebe, H.T. Shen, A survey on learning to hash, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 769–790.
[6] J. Wang, S. Kumar, S. Chang, Semi-supervised hashing for large-scale search, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2393–2406.
[7] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2009, pp. 1753–1760.
[8] C. Liu, X. Wang, T. Lu, W. Zhu, J. Sun, S.C.H. Hoi, Discrete social recommendation, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2019, pp. 208–215.
[9] Z. Zhang, Q. Wang, L. Ruan, L. Si, Preference preserving hashing for efficient recommendation, in: Proceedings of the ACM International Conference on Research on Development in Information Retrieval (SIGIR), 2014, pp. 183–192.
[10] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, T.-S. Chua, Discrete collaborative filtering, in: Proceedings of the ACM International Conference on Research on Development in Information Retrieval (SIGIR), 2016, pp. 325–334.
[11] D. Lian, R. Liu, Y. Ge, K. Zheng, X. Xie, L. Cao, Discrete content-aware matrix factorization, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2017, pp. 325–334.
[12] Y. Zhang, D. Lian, G. Yang, Discrete personalized ranking for fast collaborative filtering from implicit feedback, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 1669–1675.
[13] Y. Zhang, H. Yin, Z. Huang, X. Du, G. Yang, D. Lian, Discrete deep learning for fast content-aware recommendation, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2018, pp. 717–726.
[14] H. Liu, X. He, F. Feng, L. Nie, R. Liu, H. Zhang, Discrete factorization machines for fast feature-based recommendation, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 3449–3455.
[15] L. Ji, Q. Qin, B. Han, H. Yang, Reinforcement learning to optimize lifetime value in cold-start recommendation, in: Proceedings of the ACM International Conference on Information and Knowledge Management, 2021, pp. 782–791.
[16] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, D. Cai, Addressing the item cold-start problem by attribute-driven active learning, IEEE Trans. Knowl. Data Eng. 32 (4) (2020) 631–644.
[17] G. Hu, Y. Zhang, Q. Yang, Conet: Collaborative cross networks for cross-domain recommendation, in: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), 2018, pp. 667–676.
[18] W. Fu, Z. Peng, S. Wang, Y. Xu, J. Li, Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2019, pp. 94–101.
[19] C. Zhao, C. Li, R. Xiao, H. Deng, A. Sun, CATN: cross-domain recommendation for cold-start users via aspect transfer network, in: Proceedings of the ACM International Conference on Research on Development in Information Retrieval (SIGIR), 2020, pp. 229–238.
[20] H. Lee, J. Im, S. Jang, H. Cho, S. Chung, Melu: Meta-learned user preference estimator for cold-start recommendation, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2019, pp. 1073–1082.
[21] M. Dong, F. Yuan, L. Yao, X. Xu, L. Zhu, MAMO: memory-augmented meta-optimization for cold-start recommendation, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2020, pp. 688–697.
[22] Y. Lu, Y. Fang, C. Shi, Meta-learning on heterogeneous information networks for cold-start recommendation, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2020, pp. 1563–1573.
[23] M. Huisman, J.N. van Rijn, A. Plaat, A survey of deep meta-learning, Artif. Intell. Rev. 54 (6) (2021) 4483–4541.
[24] X. Sun, T. Shi, X. Gao, Y. Kang, G. Chen, FORM: follow the online regularized meta-leader for cold-start recommendation, in: Proceedings of the ACM International Conference on Research on Development in Information Retrieval (SIGIR), 2021, pp. 1177–1186.
[25] D. Lian, X. Xie, E. Chen, H. Xiong, Product quantized collaborative filtering, IEEE Trans. Knowl. Data Eng. 33 (9) (2020) 3284–3296.
[26] Y. Wang, H. Han, M. Guo, J. Qiao, A self-organizing deep belief network based on information relevance strategy, Neurocomputing 396 (2020) 241–253.
[27] Y. Zhang, I.W. Tsang, H. Yin, G. Yang, D. Lian, J. Li, Deep pairwise hashing for cold-start recommendation, IEEE Trans. Knowl. Data Eng. (2020).
[28] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), 2016, pp. 153–162.

[29] D. Lian, X. Xie, E. Chen, Discrete matrix factorization and extension for fast item recommendation, IEEE Trans. Knowl. Data Eng. 33 (5) (2021) 1919–1933.
[30] A. Beck, L. Tetruashvili, On the convergence of block coordinate descent type methods, SIAM J. Optim. 23 (4) (2013) 2037–2060.
[31] C. Hansen, C. Hansen, J.G. Simonsen, S. Alstrup, C. Lioma, Content-aware neural hashing for cold-start recommendation, in: Proceedings of the ACM International Conference on Research on Development in Information Retrieval (SIGIR), 2020, pp. 971–980.
[32] Y. Xu, L. Zhu, Z. Cheng, J. Li, J. Sun, Multi-feature discrete collaborative filtering for fast cold-start recommendation, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2020, pp. 270–278.
[33] M. Saveski, A. Mantrach, Item cold-start recommendations: learning local collective embeddings, in: Proceedings of the ACM Conference on Recommender systems (RecSys), 2014, pp. 89–96.
[34] K. Zhou, H. Zha, Learning binary codes for collaborative filtering, in: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2012, pp. 498–506.
[35] B. Xu, J. Bu, C. Chen, C. Wang, D. Cai, X. He, Emr: A scalable graph-based ranking model for content-based image retrieval, IEEE Trans. Knowl. Data Eng. 27 (1) (2015) 102–114.
[36] C. Zheng, L. Zhu, X. Lu, J. Li, Z. Cheng, H. Zhang, Fast discrete collaborative multi-modal hashing for large-scale multimedia retrieval, IEEE Trans. Knowl. Data Eng. 32 (11) (2019) 2171–2184.
[37] M. Weimer, A. Karatzoglou, Q.V. Le, A.J. Smola, Cofi rank-maximum margin matrix factorization for collaborative ranking, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2008, pp. 1593–1600.
[38] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, Z. Huang, From zero-shot learning to cold-start recommendation, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI), 2019, pp. 4189–4196.
[39] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, Z. Huang, Leveraging the invariant side of generative zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7402–7411.

Chengmei Peng is with the School of Information Science and Engineering, Shandong Normal University, China. Her research interests include the recommender systems and multimedia computing.

Lei Zhu is currently a professor with the School of Information Science and Engineering, Shandong Normal University. He received his B.Eng. and Ph.D. degrees from Wuhan University of Technology in 2009 and Huazhong University Science and Technology in 2015, respectively. He was a Research Fellow at the University of Queensland (2016–2017). His research interests are in the area of large-scale multimedia content analysis and retrieval. Zhu has co-/authored more than 100 peer-reviewed papers, such as SIGIR, ACM MM, IEEE TPAMI, IEEE TIP, IEEE TKDE, and ACM TOIS. His publications have attracted more than 3500 Google citations. At present, he serves as the editorial board member of 4 SCI-indexed journals, youth editorial board of IEEE/CAA Journal of Automatica Sinica. He has served as the Area Chair, Senior Program Committee or reviewer for more than 40 well-known international journals and conferences. He won ACM SIGIR 2019 Best Paper Honorable Mention Award, ADMA 2020 Best Paper Award, ACM China SIGMM Rising Star Award, Shandong Provincial Entrepreneurship Award for Returned Students, and Shandong Provincial AI Outstanding Youth Award.

Yang Xu received his Ph.D. degree at Shandong University, China, in 2018. He is currently a lecturer with the School of Information Science and Engineering, Shandong Normal University, China. His research interests are recommender systems and multimedia computing.

Yaping Li is currently a professor with the School of Information Engineering, Shandong Management University, China. Her research interests are recommender systems and multimedia computing.

Lei Guo received his Ph.D. degree in computer science from Shandong University, China, in 2015. He is currently an Associate Professor and a Master Supervisor with Shandong Normal University, China. He is a Director of Shandong Artificial Intelligence Society and a member of the Social Media Processing Committee of the Chinese Information Society. His research interests include information retrieval, social networks, and recommender systems.