Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-08705-z

Quantization-based hashing with optimal bits for efficient recommendation

Yan Zhang[1] · **Defu Liu**[1] · **Guowu Yang**[1] · **Lin Hu**[2]

Received: 10 March 2019 / Revised: 16 October 2019 / Accepted: 28 January 2020 /
Published online: 25 March 2020

Abstract
Recommendation technique has been widely applied in e-commerce systems, but the efficiency becomes challenging due to the growing scale of users and items. In recent years, several hashing-based recommendation frameworks were proposed to solve the efficiency issue successfully by representing users and items with binary codes. These hashing methods consist of two types: two-stage hashing and learning-based hashing. In this paper, we focus on putting forward to a two-stage hashing called quantization-based hashing (QBH) to alleviate the efficiency bottleneck and improve the recommendation accuracy as well. To be specific, we propose the QBH that consists of similarity quantization and norm quantization. To improve the accuracy performance, we search the optimal bits of quantization by minimizing a quantization loss function. We finally evaluate the proposed method on three public datasets to show its superiority on recommendation accuracy over other two-stage hashing methods and advantage on recommendation efficiency over the state-of-the-art recommender systems.

Keywords Recommender system · Quantization-based hashing · Recommendation efficiency · Matrix factorization

✉ Guowu Yang
guowu@uestc.edu.cn

Yan Zhang
yixianqianzy@gmail.com

Defu Liu
gxdefu@gmail.com

Lin Hu
lin.hu@ieee.org

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Qingshuihe Campus:No.2006, Xiyuan Ave, West Hi-Tech Zone, Chengdu 611731, Sichuan, China

[2] Key Laboratory of Mobile Communication, School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

1 Introduction

In the era of information explosion, information overload becomes a challenging problem, and it becomes more and more difficult to find out valuable information for the public. Personalized recommender systems, trying to match social goods with user's taste, is one of the most important and effective approaches for information overload. It has been widely applied in many fields, such as e-commerce, social network, online education systems and medical systems, and has successfully promoted the sale of products for e-commerce websites, such as Amazon [14].

Collaborative filtering (CF) has proven to be one of the most successful recommender systems. They produce top-k items that users may be interested in by exploiting the historical interaction data such as ratings, purchasing, clicking, and collection et al. Among these CF-based methods, the matrix factorization (MF) [1, 2, 6, 13, 16], have been demonstrated to achieve great success in both academia and industry. Such MF-based methods factorize an $n \times m$ user-item interaction matrix into a low-dimensional latent vector (a.k.a. feature) space where both users and items are represented by real latent vectors; then the user's preference scores for items were predicted by inner products between their real latent vectors, and the user's top-k preferred items can be produced by ranking these scores descendingly, which is called a procedure of preference prediction or the online recommendation.

Suppose there are n users and m items in a recommender system, and users/items have already been denoted as d-dimension real latent vectors, searching the top-k items that users maybe interested in is the task of online recommendation. In the real latent space, the time complexity of online recommendation is $\mathcal{O}(md + m \log k)$ [24] for a specific user u, and thus it's a critical efficiency bottleneck when the collection of items is large.

Hashing technique can provide a fast similarity search on large scale datasets, and the preference ranking in Hamming space can be also regarded as a type of similarity search. Hashing technique is significant in various applications, such as image retrieval [29–31], object tracking [9–12, 15]. Thus, hashing technique encodes users and items into binary codes in Hamming space, is a promising approach to tackle the online recommendation efficiency bottleneck. Since preference score, in this case, can be efficiently computed by bit operations, i.e., Hamming distance. One can even use a fast and accurate indexing method to find approximate top-k preferred items with sublinear or logarithmic time complexity [17, 20, 21]. Besides, each dimension of hash code can be stored by only one bit instead of 32/64 bits applied for storing one dimension data with real latent vectors, which significantly reduces the storage cost. However, many previous hashing-based recommendation improves the recommendation efficiency with the sacrifice of recommendation accuracy due to a large amount of information loss caused by a quantization procedure.

Existing hashing-based recommender systems were proposed to find some trade-off strategies between efficiency and accuracy [22, 24, 25, 27, 28]. To learn hash codes, a discrete optimization problem is proposed by minimizing a rating loss function. But solving the discrete problem often falls into an NP hard problem [4]. Many scholars tried to design strategies to get approximate solutions. These strategies consist of two types: learning-based hashing and quantization-based hashing. Learning-based hashing is often optimized via an alternating optimization for the discrete objective, and eventually reduced to solve a series of mixed-integer subproblems; thus the time cost of learning procedure is generally expensive.

Quantization-based hashing firstly solve a relaxed optimization via discarding the discrete constraints, and subsequently quantify the learned real latent vectors into binary codes by some quantization strategies.

The training procedure of quantization-based hashing is more efficient than the learning-based hashing. As introduced in the quantization-based hashing method [23], one representative work is proposed by K. Zhou et al. [28], where real latent factors are rotated and quantized into binary codes based on an iterative threshold quantization method. This method has been shown its potential in speeding up recommendation greatly, but suffers from a problem of low recommendation accuracy compared to traditional MF based models. The deterioration of recommendation performance not only depends on information loss arising from hashing itself, but also lies in not considering the change of dot product caused by binary quantization, as analyzed in Zhang et al. [27]. Thus they put forward a preference preserving hashing model (PPH), whose part of learning real latent factors place constant feature norm constraint. In the quantization process, PPH offers a method transferring real latent factors to binary codes based on similarity and norm, respectively. In their paper, most bits of each hash code came from similarity quantization. However, PPH model did not consider the accuracy loss caused by the constant norm constraint in the process of learning real latent factors. Besides, only 2 bits used in norm quantization also leads to information loss.

In this paper, we propose a quantization-based hashing framework by optimizing the number of bits to reduce information loss caused by the quantization procedure. To be specific, we propose a new quantization-based hashing framework consists of similarity quantization and norm quantization to conduct a recommendation. Thus, hash codes in this paper consist of two parts: norm based hash codes and similarity based hash codes. In order to minimize the information loss caused by the quantization procedure, we furtherly formulate a quantization loss function to learn the optimal bits (dimension) of norm quantization in Section 3.4. The dimension or the number of bits is significant for the recommendation performance, because how many bits of hash codes to some extent decides how much information carried.

Main contributions of this paper are summarized as follows:

– We put forward a principled preference preserving two-stage hashing framework called qunatization-based hashing with optimal bits (QBH), which directly preserve the user's preference presented by inner product in the real latent space.
– We propose a new quantization method composed of similarity quantization and norm quantization, which is significant to preserve the preference.
– We optimize the dimension of the norm quantization to minimize information loss caused by the quantization procedure, which is also helpful to obtain effective binary codes.
– We evaluate the proposed quantization-based hashing framework on three real-world datasets to show its superiority to competitive baselines.

We organize this paper as follows: we first introduce the background and the main idea of the proposed method in Section 1, and followed by the related works and the preliminary in Section 2. We then formulate our proposed quantization-based hashing framework in Section 3 and obtain hash codes of users and items by the corresponding quantization method. We next evaluate the proposed method on three public datasets and explain the experimental results elaborately in Section 4. Finally, we conclude this paper and propose some possible future works in Section 5.

2 Preliminary

2.1 Notation

Let user set be $U = \{1, \cdots, n\}$, and item set be $I = \{1 \cdots, m\}$. Each entry $r_{ui} \in [0, 1]$ of a rating matrix $\mathbf{R}$ present the preference of user u for item i, for rating datasets with values in other range, we can normalize them into [0, 1]. For implicit feedbacks, such as clicks, collections, etc., ratings are binary values '0' and '1', where '0' represents no interactions (or implicit feedbacks), and '1' denotes interactions between users and items. We also called '0' as the negative feedback, and '1' as the positive feedback. The observed rating $r_{ui} \geq 0$ indicate that user u has rated to item i, and the set $\Omega = (u, i)|r_{ui} \geq 0$ denotes the index set of observed ratings.

2.2 Problem formulation

Suppose n users and m items are in a recommender system, and $\mathbf{R}$ is the rating matrix stacked by observed ratings. The goal of recommender system is suggesting preferred items for users by obtained vectors of users and items. The efficiency introduced in Section 1 consists of the storage efficiency and the time efficiency from offline training and online recommendation. We will respectively introduce several main notations mentioned in this paper.

Efficiency of online recommendation: Efficiency contains storage efficiency and time efficiency. Given representations of items and users, the online recommendation also named online similarity search provides a suggestion (a subset of items) to a specific user. Specifically, suppose we have n users and m items in an online recommender system. Users and items are denoted as d-dimension vectors, respectively. Searching the top-k items that users maybe interested in is the task of online recommendation. It can be implemented by the similarity search between representations of users and items. If users and items are presented as real latent vectors, the time complexity of online recommendation is $\mathcal{O}(md + m \log k)$ for a specific user, and it is a critical efficiency bottleneck when the user's collection is large. If users and items are denoted by binary vectors (hash codes), the time cost can be fundamentally reduced to constant or sublinear complexity with the size of item's set. Besides, saving these m items by real latent vectors will cost 64 times of hash codes.

Efficiency of offline training: For the learning-based hashing, the model is trained by a discrete alternating optimization algorithm, and the discrete optimization is often bit-wise. That leads to expensive time complexity. For the quantization-based hashing, the model is first relaxed to a continuous optimization problem that can be solved by the (stochastic) gradient descending algorithm, and then we can obtain the hash codes by a quantization method with constant time complexity. Thus, the efficiency of offline training in quantization-based hashing is much less than that in learning-based hashing.

Quantization-based hashing: Given a user-item rating matrix $\mathbf{R}$, the user set U, and the item set I. For each query user $u \in U$, our goal is to recommend top-k items that user u would be interested in. Firstly, we learn binary codes by our proposed quantization-based hashing (QBH). Then, we apply the preference model to predict the user's scores for all items and recommend top-k scored items.

3 Quantization-based hashing

Differ from existing quantization-based hashing frameworks, we propose a preference preserving framework instead of similarity preserving frameworks proposed before [3, 28]. Although Zhang et al. [27] proposed a preference preserving framework, but they quantize the preference with a simple way that leads to large information loss.

Below we first introduce the preference model used in the proposed framework in Section 3.1; we then introduce the first stage of the proposed hashing framework with the matrix factorization model in Section 3.2; we finally quantize the learned real latent factors by matrix factorization in the following Section 3.3. The second stage, the quantization methodology proposed in this paper consists of similarity quantization, and norm quantization. In addition, in the norm quantization, we optimize the length of hash code to minimize information loss and find the optimal bits for the norm quantization.

3.1 Preference Model

The preference model is a key component to formulate a recommender system. The traditional preference model is usually based on Matrix Factorization [7] under a low-rank assumption. Users and items are projected into a d-dimension real latent space, and the preference of user u for item i is estimated by the inner product of real latent vectors. Similarly, the proposed hashing framework in this paper maps users and items into a d-dimension Hamming space, where preferences are estimated by Hamming distances between hash codes of users and items. In this section, we introduce how to formulate the preference model for the proposed quantization-based hashing recommendation by two different methods. The first is based on an intuitive idea of the inner product between real latent factors. The second is motivated by the state-of-the-art quantization-based hashing recommendation model.

Let real latent factor of user u and item i be represented as $\mathbf{p}_u \in \mathbb{R}^d$, and $\mathbf{q}_i \in \mathbb{R}^d$, respectively. The results of similarity quantization and the norm quantization for $\mathbf{p}_u$ are denoted as $\mathbf{p}_u^s\ \{\pm 1\}^d$ and $\mathbf{p}_u^n\ \{\pm 1\}^r$, respectively. Similarly, the quantization of $\mathbf{q}_i$ consists of two parts: the similarity quantization $\mathbf{q}_i^s\ \{\pm 1\}^d$ and the norm quantization $\mathbf{q}_i^n\ \{\pm 1\}^r$.

We denote $f(\mathbf{p}_u^s, \mathbf{q}_i^s)$ as the ratio of the same bits in hash codes $\mathbf{p}_u^s$ and $\mathbf{q}_i^s$, which also be extensively represented as the similarity between two hash codes. Similarly, $f(\mathbf{p}_u^n, \mathbf{q}_i^n)$ denotes the similarity between hash codes $\mathbf{p}_u^n$ and $\mathbf{q}_i^n$.

Owing that preference was denoted by the inner product between real latent factors of user and item in MF based recommender systems, so we present a similar preference model based on the product of hash codes respectively from norm quantization and similarity quantization. Because the inner product between two real vectors can be expressed as the product of their norms and their cosine similarity. So the first preference model of user u to item i is formulated as:

$$\hat{r}_{ui}^{(1)} = f(\mathbf{p_u^s}, \mathbf{q_i}^s) \times f(\mathbf{p_u^n}, \mathbf{q_i}^n), \tag{1}$$

where

$$f(\mathbf{p_u^s}, \mathbf{q_i}^s) = 1 - \frac{1}{d} H(\mathbf{p}_u^s, \mathbf{q}_i^s).$$

$$f(\mathbf{p_u^n}, \mathbf{q_i}^n) = 1 - \frac{1}{r} H(\mathbf{p}_u^n, \mathbf{q}_i^n).$$

Due to Hamming distance between two hash codes are the number of different bits, so we derive the following expression:

$$f(\mathbf{p_u^s}, \mathbf{q_i}^s) = \frac{1}{2} + \frac{1}{2d}\mathbf{p}_u^{s\ T}\mathbf{q}_i^s,$$
$$f(\mathbf{p_u^n}, \mathbf{q_i}^n) = \frac{1}{2} + \frac{1}{2r}\mathbf{p}_u^{n\ T}\mathbf{q}_i^n.$$

Next, we will introduce the second preference model, which is motivated by the previous state-of-the-art recommendation model, preference preserving hashing (PPH). In this model, hash codes from similarity quantization and norm quantization are concatenated into one hash code, i.e., $[\mathbf{p_u}^s, \mathbf{p_u}^n]$ denotes hash code of user u, and $[\mathbf{q_i}^s, \mathbf{q_i}^n]$ denotes hash code of item i. Similar to the first preference model, the preference is modeled as the similarity of the above unified hash codes:

$$\begin{aligned}\hat{r}_{ui}^{(2)} &= f([\mathbf{p_u}^s, \mathbf{p_u}^n], [\mathbf{q_i}^s, \mathbf{q_i}^n]) \\ &= 1 - \frac{1}{d+r}H([\mathbf{p_u}^s, \mathbf{p_u}^n], [\mathbf{q_i}^s, \mathbf{q_i}^n]) \\ &= 1 - \frac{1}{d+r}\big(H(\mathbf{p_u}^s, \mathbf{q_i}^s) + H(\mathbf{p_u}^n, \mathbf{q_i}^n)\big) \\ &= \frac{1}{2} + \frac{1}{2(d+r)}(\mathbf{p}_u^{s\ T}\mathbf{q}_i^s + \mathbf{p}_u^{n\ T}\mathbf{q}_i^n) \end{aligned} \quad (2)$$

The second preference model is similar to PPH model, because they both straightly concatenates hash codes from similarity quantization and norm quantization. But they differ in the norm quantization and the constraints of binary codes. In addition, PPH quantizes norms with only two bits, while the proposed QBH quantizes norms with an optimal number of bits, which will be introduced in Section 3.4.

The similarity quantization and norm quantization will be separately introduced in Sections 3.3.1 and 3.3.2. The above two types of preference models will be applied in our quantization-based hashing recommendation framework, respectively. Although they are different, but they both aim to achieve the same goal – preserving users' preference.

3.2 Matrix factorization

In order to better characterize items and users by known ratings or implicit feedbacks. Matrix factorization learn real latent factors by minimizing the square loss of observed ratings and the predicted ratings. In order to avoid overfitting, two Frobenius norm regularizer terms of real latent factors $\mathbf{P}$ and $\mathbf{Q}$ are added [8], and the objective represented as follows:

$$\begin{aligned}&\underset{\mathbf{P},\mathbf{Q}}{\arg\min} \sum_{(u,i)\in\Omega} c_{ui}(r_{ui} - \mathbf{p}_u^T\mathbf{q}_i)^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) \\ &\text{s.t.}\mathbf{P} \in \mathbb{R}^{n\times d}, \mathbf{Q} \in \mathbb{R}^{m\times d}, \end{aligned} \quad (3)$$

where $\mathbf{P}$, $\mathbf{Q}$ are real latent matrices stacked by real latent factors of users U and items I, respectively. Ω is the index set of observed ratings in $\mathbf{R}$. λ is the regularizer hyper-parameter. c_{ui} is the confidence value when we transform the explicit ratings to implicit feedbacks, thus $c_{ui} = 1$ if $r_{ui} > 0$; otherwise, $c_{ui} = 0$. So the matrix factorization becomes weighted matrix factorization (WMF) [5]. Thus the proposed method can be applied for explicit ratings implicit feedbacks; while the previous quantization-based hashing methods NQ [23] and DPH [25] can only applied for explicit ratings.

Generally, $\mathbf{R}$ is a highly sparse matrix in real world datasets. This model aims to learn real latent matrices $\mathbf{P}$ and $\mathbf{Q}$ that can be furtherly applied to reconstruct users' ratings. To solve the above problem, we choose stochastic gradient decent method to learn real latent factors $\mathbf{p}_u$ and $\mathbf{q}_i$. Then we can obtain hash codes of users and items from the learned real latent factors by the proposed quantization-based hashing in the following section.

3.3 Quantization-based hashing

Quantization-based hashing presented in this paper consists of similarity quantization and norm quantization, and thus hash codes of users and items are composed of the corresponding two parts.

3.3.1 Similarity quantization

Similarity quantization aims to quantize the cosine similarity of two real latent factors. Suppose hash codes of $\mathbf{P}$ and $\mathbf{Q}$ transferred by similarity quantization are represented as $\mathbf{P^s} \in \{\pm 1\}^{d\times n}$ and $\mathbf{Q^s} \in \{\pm 1\}^{d\times m}$, respectively, and $\mathbf{p_u}^s \in \mathbf{P^s}$, $\mathbf{q_i}^s \in \mathbf{Q^s}$.

As discussed in Section 3.1, the predicted preference based on real latent factors' inner product consists of two parts: cosine similarity and norms. Intuitively, if we can separately design a ranking preserving hash function for cosine similarity and norms, then the items' ranking in Hamming space will be consistent with that in real space, and hence the preference can be preserved. In this paper, we aim to design a ranking preserved hash function $H : \mathbb{R} \to \{\pm 1\}^d$ which maps any real number in $\mathbb{R}$ to a specific hash codes in $\{\pm 1\}^d$.

For example, let $a,\ b$ be in $\mathbb{R}$, the corresponding hash codes via hash function H are $\mathbf{a^h},\ \mathbf{b^h} \in \{\pm 1\}^d$, respectively. We define the query vector in $\{\pm 1\}^d$ as $\mathbf{q} = [1, 1, \cdots, 1]_d{}^T$. If $a \geq b$, the corresponding similarities between $\mathbf{a^h},\ \mathbf{b^h},\ \mathbf{q}$ satisfy

$$f(\mathbf{a^h}, \mathbf{q}) \geq f(\mathbf{b^h}, \mathbf{q}),$$

then H is called as ranking preserved hash function.

Based on the above discussion, for each user u we denote H_u^s as the ranking preserved hash function. The query vector $\mathbf{p}_u^s = [1, 1, \cdots, 1]_d{}^T$ is regarded as the similarity based hash codes of $\mathbf{p}_u$. We define H_u^s as

$$\begin{aligned} H_u^s &: \mathbb{R} \to \{\pm 1\}^d, \\ \cos\theta_{\mathbf{p}_u,\mathbf{q}_i} &\to \mathbf{q}_i^s = [\mathbf{q}_i^s(1), \mathbf{q}_i^s(2), \cdots, \mathbf{q}_i^s(d)]^T, \end{aligned} \tag{4}$$

where

$$\mathbf{q}_i^s(m) = \begin{cases} 1\ , & if\ sgn(\mathbf{p}_u(k)) = sgn(\mathbf{q}_i(k)), \\ -1\ , & else, \end{cases}$$

and $k \in \{1, 2, \cdots, d\}$.

We get similarity based hash codes of items and users by the above ranking preserved hash function H_u^s with low time complexity. Since we only need to compare signs of two real latent factors. Furthermore, the query vector will search similar vectors with $\mathbf{p}_u$, since the query hash code $\mathbf{p}_u^s$ can be regarded as a metric of similarities between user u and all items. If $f(\mathbf{p}_u^s, \mathbf{q}_i^s) \geq f(\mathbf{p}_u^s, \mathbf{q}_j^s)$, then item i is considered as more similar with user u than item j.

3.3.2 Norm quantization

In this section we propose a norm quantization method to generate hash codes from norms of real latent factors, that can improve recommendation accuracy to a great extent.

As discussed in Section 3.3.1, in order to preserve items' ranking, we have designed a ranking preserved hash function based on cosine similarity. In the norm quantization, we also design a ranking preserved hash function H_u^n as the norm based hash function for user u. Similarly, the query vector is designed as $\mathbf{p}_u^n = [1, 1, \cdots, 1]_r{}^T$, which can be also regarded as the norm based hash codes of $\mathbf{p}_u$, and we define H_u^n as

$$\begin{aligned} H_u^n &: \mathbb{R} \rightarrow \{\pm 1\}^r, \\ \|\mathbf{q}_i\| &\rightarrow \mathbf{q}_i^n = [\mathbf{q}_i^n(1), \mathbf{q}_i^n(2), \cdots, \mathbf{q}_i^n(r)]^T, \end{aligned} \tag{5}$$

where

$$\mathbf{q}_i^n(t) = \begin{cases} 1 \ , & if\ \|\mathbf{q}_i\| \geq \frac{10t-5}{2r}, \\ -1, & else, \end{cases}$$

$t \in \{1, 2, \cdots, r\}$, r is the number of bits in the norm based hash code. The value of r will be determined in the following section. Our method is designed under an assumption that all norms are within [0,1]. It is an appropriate assumption because the norms of $\mathbf{p}_u$ and $\mathbf{q}_i$ are restricted by L_2-norm regularizer terms in (3). Therefore, most of the learned real latent factors, $\mathbf{p}_u$ and $\mathbf{q}_i$, have small norms. Besides, for norms greater than 1 will be transferred to $[1, 1, \cdots, 1]_r{}^T$ by the above norm based hash function, so the norm quantization method does also work.

Through the above norm based hash function, we can obtain $r-$bit norm-based hash codes that can achieve higher accuracy than 2-bit norm-based hash codes in PPH, because r-bit quantization is more refined. Besides, the query vector $\mathbf{p}_u^n$ can be used to search the most similar items for user u.

The above norm-based hash function is also ranking preserved. For any user u and two items i and j, let the corresponding real latent factors be denoted as $\mathbf{p}_u$, $\mathbf{q}_i$, and $\mathbf{q}_j$, respectively. If $\|\mathbf{q}_i\| \geq \|\mathbf{q}_j\|$. From (4) and (5), we attain $\mathbf{q}_i^n$ is more similar with $\mathbf{p}_u^n$ than $\mathbf{q}_j^n$, i.e.,

$$sim(\mathbf{p}_u^n, \mathbf{q}_i^n) \geq sim(\mathbf{p}_u^n, \mathbf{q}_j^n).$$

Hence, the norm based function is also ranking preserved.

For example, for a particular user u and two items i and j, let the corresponding real latent factors be $\mathbf{p}_u$, $\mathbf{q}_i$, and $\mathbf{q}_j$, respectively, $\|\mathbf{p}_u\| = 2.6$, $\|\mathbf{q}_i\| = 2.2$ and $\|\mathbf{q}_j\| = 3.7$. Suppose $r = 5$ (which will be determined in the Section 3.4), then we calculate the norm based hash codes by (4),

$$\begin{aligned} \mathbf{p}_u^n &= [1,\ 1,\ 1,\ 1,\ 1], \\ \mathbf{q}_i^n &= [1,\ 1,\ 1,\ -1,\ -1], \\ \mathbf{q}_j^n &= [1,\ 1,\ 1,\ 1,\ -1]. \end{aligned}$$

Due to $\|\mathbf{q}_i\| \leq \|\mathbf{q}_j\|$, and the similarity of the two hash codes meet

$$sim(\mathbf{p}_u^n, \mathbf{q}_i^n) \leq sim(\mathbf{p}_u^n, \mathbf{q}_j^n).$$

Therefore, the above norm based hashing function is ranking preserved.

3.4 Optimal bits of QBH

The similarity quantization usually leads to significant information loss. To compensate the loss, we propose a new norm quantization method in Section 3.3.2. In this section, we will discuss how many bits should we apply for norm quantization. Based on hash codes from similarity quantization, we propose a model to learn the optimal bits by minimizing a quantization loss function. To our knowledge, it is the first work to learn the optimal dimension of hash codes. By substituting the learned optimized bits r into the norm quantization, the information loss can be reduced significantly.

As similarity-based hashing function and norm-based hashing function are both ranking preserved. Hash codes in this paper consists of two parts respectively from similarity quantization and norm quantization. We estimate users' preferences in Hamming space according to (1) and (2), separately. To minimize the information loss caused by quantization, we learn the optimal dimension r of norm quantization by minimizing the following loss

$$\arg\min_{r} \sum_{(u,i)\in\Omega} (\mathbf{p}_u^T \mathbf{q}_i - \hat{r}_{ui})^2 + \alpha r, \tag{6}$$

where $\hat{r}_{ui}$ presents the predicted preference by hash code. To avoid overfitting caused by long hash codes and reduce the complexity of quantization, we add the second term as a regularizer term of r, and α, β are hyper-parameters selected by cross validation. By substituting preference models proposed in Section 3.1 into the objective 6, (6) can be rewritten as:

$$\arg\min_{r} \sum_{(u,i)\in\Omega} \left(\mathbf{p}_u^T \mathbf{q}_i - \frac{1}{4}(1 + \frac{1}{r}\mathbf{p}_u^{n\,T}\mathbf{q}_i^n)\cdot(1 + \frac{1}{d}\mathbf{p}_u^{s\,T}\mathbf{q}_i^s)\right)^2 + \alpha r, \tag{7}$$

and

$$\arg\min_{r} \sum_{(u,i)\in\Omega} \left(\mathbf{p}_u^T \mathbf{q}_i - \frac{1}{2} + \frac{1}{2(d+r)}(\mathbf{p}_u^{s\,T}\mathbf{q}_i^s + \mathbf{p}_u^{n\,T}\mathbf{q}_i^n)\right)^2 + \beta r. \tag{8}$$

We can get the optimal r by an iterative search method. The above problems (7) and (8) can be solved by Algorithm 1.

Algorithm 1 Learning optimized r-bit of norm based hash codes.

```
Input: Real latent matrices $\mathbf{P}$, $\mathbf{Q}$, and similarity based hash codes $\mathbf{P^s}$, $\mathbf{Q^s}$, and the
        hyper-parameters $\alpha$, $\beta$;
Output: The optimal dimension $r$ of norm quantization, norm based hash codes $\mathbf{P^n}$,
        $\mathbf{Q^n}$;
for $r = 1$ to 100 do
    Compute norm-based hash codes $\mathbf{q}_u^n$, $\mathbf{q}_i^n$ from (5) ;
    Compute the objective of (7) or (8) ;
return $r$, which minimize the objective (7) or (8)
    $\mathbf{P^n}$ and $\mathbf{Q^n}$, can be computed by (5)
```

We first obtain the similarity-based hash codes $\mathbf{P}^s$ and $\mathbf{Q^s}$ by (4), and we then choose the optimal hyper-parameters α, β, and λ by grid search within $[10^{-3}, 10^3]$ on validation datasets. For the iteration search from 1 to 100, we choose the optimal r, which minimize the objective (7) or (8). Thus, the r-dimension norm quantization can minimize the information loss caused by quantization procedure.

The complexity of the proposed Algorithm 1 is much less than learning-based hashing methods. Because in QBH, the similarity quantization costs $\mathcal{O}(d)$ and norm quantization costs $\mathcal{O}(r)$. The iteration loop of the Algorithm 1 costs $\mathcal{O}(100(d+r))$. Thus the total complexity of the quantization-based hashing is $\mathcal{O}(maxd, r)$. Thus The overall complexity of the proposed QBH is $\mathcal{O}(max(m+n)d, (m+n)r)$. However, the complexity of the learning-based hashing DCF [22] is $\mathcal{O}(d^2(m+n+x))$, in addition, the complexity of the learning-based DRMF [26] is $\mathcal{O}\big(\|\mathbf{R}\|_0 d^2 + (m+n)(d^{3.5})\big)$. Thus the proposed QBH has evident advantage over the learning-based hashing methods in the time complexity of the training procedure.

4 Experiment

In this section, we first introduce three real world datasets and experimental setup. Then, we introduce the evaluation metric NDCG@k and four competing baselines appeared in the experiments. Finally, we analyze experimental results of the recommendation accuracy under NDCG@10 compared with three quantization-based hashing frameworks and the state-of-the-art matrix factorization with L_2 norm regularizer terms. Experiments on three real world datasets show that our quantization scheme can achieve much higher recommendation accuracy than previous quantization methods.

4.1 Datasets

We evaluate our method on one IMDB dataset [1] and two MovieLens datasets [2]: IMDB-1M, MovieLens-1M and MovieLens-10M. IMDB-1M data contains 712,772 ratings from 6040 users to 2738 movies, which was also used by H. Shan [19]. MovieLens-1M contains 1,000,209 ratings from 6040 users to 3706 movies. MovieLens-10M includes 10,000,054 ratings from 69878 users to 10677 movies. All ratings are within [1,5], and users were selected at random for inclusion. All users selected had rated at least 20 movies.

4.2 Data Pre-processing

At first, we normalize all ratings into the interval of [0,1] to keep consistent with the predicted preference defined in this paper. Then we divide each of the above three datasets into 3 disjoint subsets randomly: training dataset, validation dataset and testing dataset. For each user, the training datasets consists of 20 ratings, and the validation datasets includes 10 ratings, and the rest at least 10 ratings makes up the testing datasets, because we need to predict top-10 possible interesting items for each user. Thus, users having less than 40 ratings are not considered in our experiments.

Training datasets and validation datasets are applied to learn real latent factors by solving the problem in (3) via Stochastic Gradient Descending (SGD). Validation datasets are used to select the regularizer parameter λ in (3), and the regularizer parameter in Equation (7) and (8). Experiments show that the recommendation accuracy on validation datasets is quite stable within a large range of λ round the optimal one. Thus we choose λ=10 for

[1] https://www.imdb.com

[2] https://grouplens.org/datasets/movielens

all datasets. Testing datasets are applied to test the recommendation performance on the evaluation metric of the proposed method.

4.3 Evaluation methods

As a matter of fact, recommendation can also be regarded as items' ranking for each user. As introduced in Section 1, the goal of recommendation is to find out the top-k items that users may be interested in. We choose NDCG@k [18] as the evaluate metric to measure the recommendation performance, which has been widely used to evaluate the quantity of a ranking. The NDCG is the normalized Discounted cumulative gain (DCG), which can be looked as the similarity of the predicted k items' ranking to the ideal ranking (the true ranking). The NDCG@k takes vales within [0, 1]. If NDCG@k is equal to 1, which means the predicted ranking is totally equal to the true ranking; on the contrary, NDCG@k equals to 0 means the predicted ranking is totally different from the true ranking.

4.4 Comparison methods

In this section, we introduce four stat-of-the-art recommendation frameworks. Three of them are quantization-based hashing methods and one of them is the most successful recommendation framework – Matrix Factorization (MF).

MedThresh: In 2012, K. Zhou et al. proposed a quantization-based hashing method for collaborative filtering [28], which quantize the learned real latent factors by a median threshold function (MedThresh). In their paper, they named the method as 'CFCodeReg' or 'CFCodePair'.

ITQ: Iterative Quantization (ITQ) [3] was proposed to learn similarity-preserving binary codes for efficient similarity search in large-scale image collections. The authors formulated this problem in terms of finding a rotation of zero-centered data so as to minimize the quantization error of mapping this data to the vertices of a zero-centered binary hypercube, and proposed a simple and efficient alternating minimization algorithm to accomplish this task.

PPH: This is a two-stage Preference Preserving Hashing framework (PPH) [27]. Different from traditional MF (Matrix Factorization model), they emphasize the important of real latent feature norm. They imposed Constant Feature Norm (CFN) constraint when learning user/item latent representation, and then quantized their magnitudes and similarity respectively. PPH quantized each real latent vector into $(r-2)$-bit phase codes and $2-$bit magnitude codes. Hence, in order to keep the code length consistent to our framework, we only learned $(r-2)$-dim real latent features at the relaxed optimization stage.

MF-L2: Matrix Factorization [8] is the state-of-the-art real-valued collaborative filtering. It maps both users and items to a joint latent factor space such that the rating of a user and an item can be predicted by calculating inner products of their vectors. As discussed in Section 1, MF can achieve high recommendation accuracy, but online recommendation with large datasets is a challenge in real latent space.

4.5 Experimental results

In this section, we evaluate the effectiveness of the proposed quantization-based hashing framework on three public datasets. Experimental shows the superiority of QBH$_1$ and QBH$_2$

which preference models are respectively defined in (1) and (2). Despite that they use the same quantization method to obtain hash codes, but they have different preference model, which leads to different performance in recommendation. In the following, we will analyze the experimental results in detail.

Figure 1 (**Left.**) indicates the proposed hashing-based recommendation has the efficiency advantage over real-valued recommendation methods for online recommendation. When training finished, online recommendation is conducted in Hamming space for hashing-based recommendation (QBH proposed in this paper); while for real-valued recommendation, onine recommendation is conducted in real latent space. From the figure, we can conclude the time cost of Hamming rank in Hamming space is much less that that in real latent space.

Figure 1 (**Right.**) show the recommendation accuracy (NDCG@10) of the proposed QBH_1 and QBH_2 and four competing baselines on IMDB-1M, and it shows the NDCG@10 value varies with the dimension of hash codes. From this figure, we can conclude that QBH_1 and QBH_2 provide more accurate recommendation on IMDB-1M, and QBH_1 is more robust with the bits (dimension) of hash codes than QBH_2. Two classic quantization-based hashing frameworks ITQ and MedThresh are stable with the dimension of hash codes, but the recommendation accuracy is not as good as our proposed methods. MF-L2 can provide the highest accurate recommendation, because it learned the real latent factors for users and items, and real latent vectors intuitively carries much more information than binary hash codes. But MF-L2 came across the efficiency problem when conducted online recommendation. Thus, hashing frameworks have the advantage of online recommendation efficiency over real-valued recommendation.

We separately analyze the recommendation efficiency from the perspective of time cost and storage cost. For the time cost, Suppose users and items have been denoted as $d-$dimension real latent factors and $d-$dimension hash codes, respectively. For a specific user u, the time cost of real-valued methods for recommending top$-k$ items from m items is $\mathcal{O}(md + m\log k)$; while searching nearest neighbors in $d-$dimension Hamming space is extremely fast. There are two methods to search the top-k items: one is Hamming ranking, it can be conducted by ranking Hamming distances with the query hash code (user), it has the complexity of $\mathcal{O}(m)$, which is linear with the item data size. The other one is hashing lookup. Similar hash codes are searched in a hamming ball centered at the query hash code. The time complexity is independent of the item data size. Therefore, Hashing based

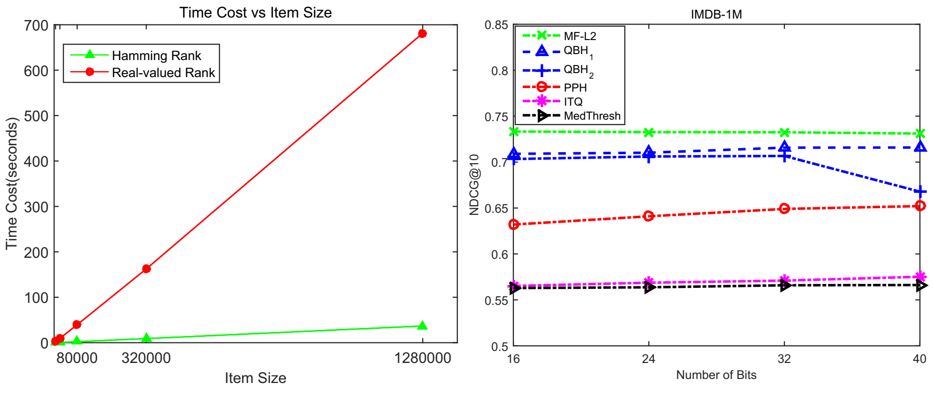


Fig. 1 **Left.** Efficiency comparison between QBH and real-valued recommendation methods. **Right.** Recommendation performance(NDCG@10) on IMDB-1M

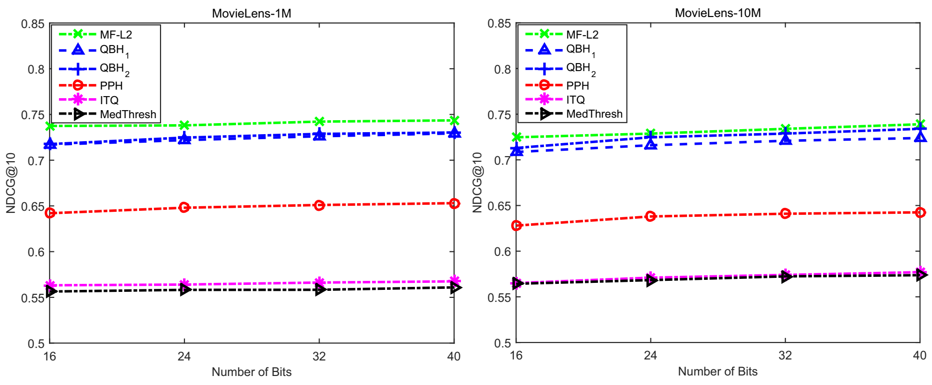


Fig. 2 Recommendation performance(NDCG@10) on MovieLens-1M and MovieLesn-10M

recommendation has evident superiority over the real-valued recommendation. For the storage cost, in real latent space, each dimension of a vector are saved at least 64 bits. With the dimension becoming large, it will cost much more space; while in Hamming space, only one bit is needed to store one dimension of a hash code. Thus the storage cost is reduced significantly.

Figure 2 shows the recommendation accuracy (NDCG@10) comparison on two MovieLens datasets. On MovieLens-1M, the performance of QBH_1 is very similar to QBH_2. The recommendation accuracy of QBH_1 and QBH_2 is very close the real-valued matrix factorization method. Meanwhile, they both provide better recommendation than other three quantization-based hashing methods PPH, ITQ, and MedThresh. Which means the proposed QBH provides a better quantization method than the previous quantization methods.

On MovieLens-10M, the results of four baselines are very similar to MovieLens-1M except the performance of QBH_1 and QBH_2. Differ from the performance on IMDB-1M, QBH_2 provides more accurate than QBH_1 on MovieLens-1M, which means QBH_1 or QBH_2 is not always a better way or a worse way to model the preference. In some cases, QBH_1 can provide a better recommendation than QBH_2; in other cases, QBH_2 performs better than QBH_1. But compared with other hashing frameworks, our proposed QBH_1 and

Table 1 The optimal number of bits for norm quantization given $d = 24$ on IMDB-1M

r	QBH_1	QBH_2	$loss_1$	$loss_2$
5	0.6786	0.6752	553.9470	1912.8569
10	0.7071	0.7026	495.5862	1792.6490
15	0.7163	0.7129	493.4218	1711.2350
20	0.7275	0.7238	490.8735	1652.2579
25	0.7295	0.7257	$\boxed{490.8448}$	1612.2169
30	0.7273	0.7259	491.1291	1562.5105
35	0.7298	0.7276	495.5513	1550.2099
40	0.7284	0.7283	499.6741	$\boxed{1546.3874}$
45	0.7274	0.7278	502.4244	1547.6201
50	0.7291	0.7286	504.5943	1556.5320

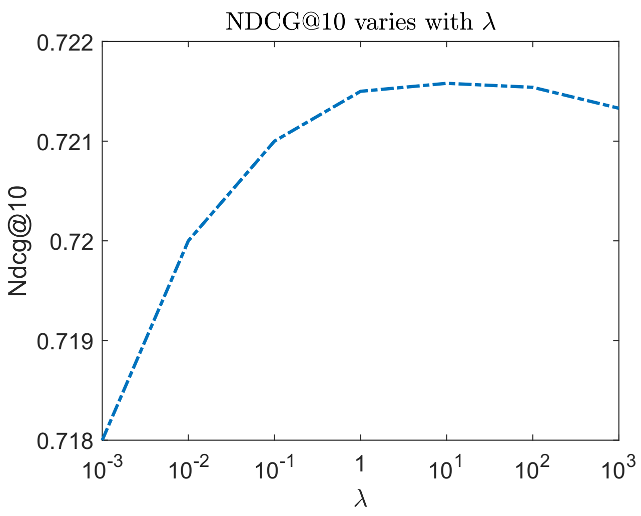


Fig. 3 Recommendation accuracy varies with λ

QBH_2 can provide much higher recommendation. That can be attributed to the refined norm quantization and the optimizing bits algorithm in the proposed QBH framework.

The optimal bits of the norm quantization varies with the dimension d of real latent factors and datasets. In Table 1, we list the recommendation accuracy (NDCG@10) of QBH_1 and QBH_2 variation with the dimension r on IMDB-1M. In the experiment, we set the hyper parameters $\alpha = 0.7$ and $\beta = 10$, and the dimension of real latent factors $d = 24$.

Table 1 tells us the optimal bits of the norm quantization r for two different QBH methods are different. The optimal bits r of QBH_1 is 25, while the optimal bits of QBH_2 is 40, because the corresponding loss function reach the minimum value. Despite that the optimal bits are different, they have the similar recommendation performance. Besides, the recommendation accuracy at $r = 35$ is a little bit greater than that at $r = 25$, but the loss function expressed in (7) is minimized at $r = 25$, thus we set the optimal bits of norm quantization for QBH_1 as 25 instead of 35. Similarly, for QBH_2, we set the optimal bits as 40 instead of 50. In fact, the increasing bits leads to low efficiency in online recommendation, and also leads to high model complexity. So it's reasonable to add a regularizer term of r in the two loss functions, which helps find the optimal bits for norm quantization, and meanwhile decreases the model complexity.

We set the λ by the grid search from $[10^{-3}, 10^{3}]$. From the Fig. 3, we can conclude that the recommendation performance is quite stable with a large range of λ around the optimal value. Thus in our paper we set $\lambda = 10$.

5 Conclusion

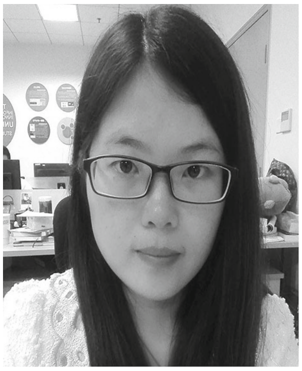
In this paper, we propose two new quantization-based hashing frameworks to preserve users' preference, and they share the same quantization method, but they are designed by different preference models. Specifically, for QBH_1, the preference is modeled as the product of the norm similarity and cosine similarity in Hamming space; for QBH_2, we model the preference as the similarity between hash codes stacked by results of norm quantization and cosine similarity quantization. Then, to learn the optimal bits of norm quantization, we design an objective by minimizing the difference between the predicted preference in real latent space and in Hamming space, and add a regularizer term on the objectives to avoid overfitting. Finally, we evaluate the effectiveness of QBH regarding NDCG@k on three public datasets to show its consistent superiority over the competing quantization-based hashing baselines and the real-valued recommendation method.

Acknowledgements This work is supported by National Natural Science Foundation of China (Grant No. 61572109, 61801060).

References

1. Baltrunas L, Ludwig B, Ricci F (2011) Matrix factorization techniques for context aware recommendation. In: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, pp 301–304
2. Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. Aaai 12:17–23
3. Gong Y, Lazebnik S, Gordo A, Perronnin F (2013) Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans Pattern Anal Mach Intell 35(12):2916–2929
4. Håstad J (2001) Some optimal inapproximability results. J ACM (JACM) 48(4):798–859
5. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, Ieee, pp 263–272
6. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, pp 135–142
7. Koren Y, Bell R (2011) Advances in collaborative filtering. In: Recommender systems handbook, Springer, pp 145–186
8. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
9. Lan X, Zhang S, Yuen PC, Chellappa R (2017) Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker. IEEE Trans Image Process 27(4):2022–2037
10. Lan X, Ye M, Zhang S, Zhou H, Yuen PC (2018) Modality-correlation-aware sparse representation for rgb-infrared object tracking. Pattern Recognition Letters
11. Lan X, Ye M, Shao R, Zhong B, Jain DK, Zhou H (2019) Online non-negative multi-modality feature template learning for rgb-assisted infrared tracking. IEEE Access 7:67761–67771
12. Lan X, Ye M, Shao R, Zhong B, Yuen PC, Zhou H (2019) Learning modality-consistency feature templates: a robust rgb-infrared tracking system. IEEE Transactions on Industrial Electronics
13. Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 831–840
14. Linden G, Smith B, York J (2003) Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE 7(1):76–80
15. Ma C, Liu C, Peng F, Liu J (2016) Multi-feature hashing tracking. Pattern Recogn Lett 69:62–71
16. Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, pp 931–940
17. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: Proceedings of VISAPP'09, pp 331–340
18. Qi GJ, Hua XS, Zhang HJ (2009) Learning semantic distance from community-tagged media collection. In: Proc ACM MM, pp 243–252
19. Shan H, Banerjee A (2010) Generalized probabilistic matrix factorizations for collaborative filtering. In: Proc IEEE ICDM, pp 1025–1030
20. Wang J, Kumar S, Chang SF (2012) Semi-supervised hashing for large-scale search. IEEE Trans Pattern Anal Mach Intell 34(12):2393–2406

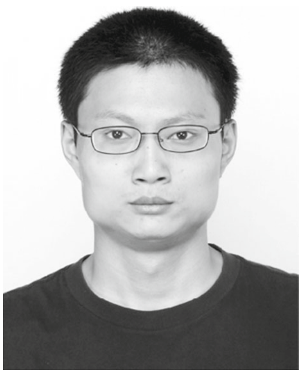
21. Wang J, Shen HT, Song J, Ji J (2014) Hashing for similarity search: A survey. arXiv preprint arXiv:14082927
22. Zhang H, Shen F, Liu W, He X, Luan H, Chua TS (2016) Discrete collaborative filtering. In: Proceedings of SIGIR'16, vol 16
23. Zhang Y, Yang G, Lian D, Wen H, Wu J (2016) Constraint free preference preserving hashing for fast recommendation. In: 2016 IEEE Global Communications Conference, GLOBECOM, IEEE, pp 1–6
24. Zhang Y, Lian D, Yang G (2017) Discrete personalized ranking for fast collaborative filtering from implicit feedback. In: AAAI, pp 1669–1675
25. Zhang Y, Yang G, Hu L, Wen H, Wu J (2017) Dot-product based preference preserved hashing for fast collaborative filtering. In: 2017 IEEE International Conference on Communications (ICC), IEEE, pp 1–6
26. Zhang Y, Wang H, Lian D, Tsang IW, Yin H, Yang G (2018) Discrete ranking-based matrix factorization with self-paced learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, pp 2758–2767
27. Zhang Z, Wang Q, Ruan L, Si L (2014) Preference preserving hashing for efficient recommendation. In: Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, pp 183–192
28. Zhou K, Zha H (2012) Learning binary codes for collaborative filtering. In: Proc ACM SIGKDD, pp 498–506
29. Zhu L, Shen J, Xie L, Cheng Z (2016) Unsupervised visual hashing with semantic assistant for content-based image retrieval. IEEE Trans Knowl Data Eng 29(2):472–486
30. Zhu L, Huang Z, Liu X, He X, Sun J, Zhou X (2017) Discrete multimodal hashing with canonical views for robust mobile landmark search. IEEE Trans Multimedia 19(9):2066–2079
31. Zhu L, Huang Z, Li Z, Xie L, Shen HT (2018) Exploring auxiliary context: discrete semantic transfer hashing for scalable image retrieval. IEEE Trans Neural Netw Learn Syst 29(11):5264–5276



Yan Zhang received her B.S. degree in mathematics and applied mathematics from Sichuan normal university, China. She is currently working toward the Ph.D. degree in School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Her research interests include recommender system, transfer learnig, machine learning, and logic synthesis.

Defu Liu is currently pursuing a Ph.D. at the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. His research interests include verification, quantum computing and machine learning.

Guowu Yang received his B.S. degree from University of Science and Technology of China in 1989, received his M.S. degree from Wuhan University of Technology in 1994, and received his Ph.D. degree in electrical and computer engineering from Portland State University in 2005. He has worked at Wuhan University of Technology from 1989 to 2001, at Portland State University from 2005 to 2006. He is a full professor at University of Electronic Science and Technology of China now. His research interests include verification, logic synthesis, quantum computing and machine learning. He has published over 100 journal and conference papers.



Lin Hu is currently a lecturer with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. He received Ph.D. degree in communication and information systems from the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China in 2017. His research interests include cooperative communication systems, convex optimization for wireless communication and signal processing, and physical layer security.