Constraint Free Preference Preserving Hashing for Fast Recommendation

Conference Paper · December 2016

DOI: 10.1109/GLOCOM.2016.7841687

5 authors, including:

Hong Wen
University of Texas MD Anderson Cancer Center
130 PUBLICATIONS **1,452** CITATIONS
SEE PROFILE

Jinsong Wu
University of Chile
165 PUBLICATIONS **1,703** CITATIONS
SEE PROFILE

Constraint Free Preference Preserving Hashing for Fast Recommendation

Yan Zhang, Guowu Yang, Defu Lian†, Hong Wen⋆, Jinsong Wu‡
Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, 611731, China
†Big Data Research Center, University of Electronic Science and Technology of China, Chengdu, 611731, China
⋆National Key Lab of Commun., University of Electronic Science and Technology of China, Chengdu, 611731, China
‡Department of Electrical Engineering, Universidad de Chile, Santiago, Chile
Emails: yixianqianzy@gmail.com, guowu@uestc.edu.cn, dove@uestc.edu.cn, sunlike@uestc.edu.cn, wujs@ieee.org

***Abstract*—Recommender systems have been widely used to deal with information overload, by suggesting relevant items that match users' personal interest. One of the most popular recommendation techniques is matrix factorization (MF). The inner products of learned latent factors between users and items can estimate users' preferences for items with high accuracy, but the preferences ranking is time consuming. Thus, hashing-based fast search technologies were exploited in recommender systems. However, most previous approaches consist of two stages: continuous latent factor learning and binary quantization, but they didn't well deal with the change of inner product arising from quantization. To this end, in this paper, we propose a constraint free preference preserving hashing method, which quantizes both norm and similarity in dot product. We also design an algorithm to optimize the bit length for norm quantization. The performance of our method is evaluated on three real world datasets. The results confirm that the proposed model can improve recommendation performance by 11%-15%, as compared with the state-of-the-art hashing approaches.**

***Index Terms*—Recommender system, preference ranking, norm quantization, optimal bit.**

I. Introduction

In the era of information explosion, information overload becomes a challenging problem, thus it is more and more difficult to find out valuable information for everyone. Personalized recommender system, trying to match social goods with user 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 Amazon [1].

One of the most effective personalized recommendation models is based on matrix factorization (MF), where both users and items are projected into the same joint latent space. A particular user's preference over an item is then based on their dot product in the latent space. In Netflix competition, several variants of MF were applied for movie recommendation [2], demonstrating that MF-based models are superior to neighbor-based recommendation techniques. However, due to large size of item set, ranking items based on the inner product preference for all users is time-consuming. The total time complexity is $\mathcal{O}(MN)$, where M and N are the number of items and users in a dataset, respectively. On the entire set of Netflix dataset, such preferences ranking takes several days [3]. Generally speaking, the real world datasets are very large, even larger than the Netflix dataset (100-million ratings). Therefore, it will be much more expensive for preference ranking in the real world applications.

In order to speed up recommendation, some technologies have been proposed for efficient recommendation [4]–[6]. Among them, hashing is one of the most efficient technologies for some applications due to its low memory cost and fast query speed. Hashing recommendation techniques quantize real latent factors to binary latent factors, the similarity computation between two binary latent factors is much faster than that with real ones. They can achieve low time complexity that is almost irrelevant to the size of item space [3]. Besides, the storage requirement is generally lower than the need of traditional recommender system.

Most previous methods consist of two stages, one of which is to learn real latent factors for users and items based on stochastic gradient descent or alternative least square, and the other of which is to quantize latent factors in a binary way. One representative work is proposed by K. Zhou et al. [7], where latent factors are rotated and quantized into hash code based on iterative threshold quantization. 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 change of dot product caused by binary quantization, as analyzed in Zhang et al. [3]. 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 code came from similarity quantization. Norms were quantized by only two bits in order to compensate the accuracy loss caused by 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-bit of norm based hash codes will also lead to information loss.

Existing hashing recommendation technologies suffered

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

from low accuracy owing to the fact that much information had been lost in the process of quantization. They often generated hash codes based on the similarities between real latent factors of users and items, such as [7], [8]. However, recommender systems focus on users' preferences over items instead of their similarities. Besides, the accuracy of hashing recommendation is generally influenced by the bit of hash codes. But, there is little study on the optimization bit length for hash codes. Previous hashing recommendation methods often generate the same bit of hash codes with that of real latent factors, which lead to a sub-optimal solution from the perspective of space and time.

To overcome drawbacks mentioned above, we propose a constraint free preference preserving hashing method that keeps users' preferences ranking over items in this paper. We first learn the real latent factors for users and items based on a L_2 regularized MF model but without constant feature norm constraint as PPH, which will be introduced in section II. We then transfer the real latent factors into hash codes based on cosine similarities between them. But normally, the similarity based hash codes will lead to much information loss. In MF, users' preferences are evaluated by inner products of the real features of users and items. As we know, inner products consist of norms and cosine values. If we only quantize the cosine values, it's no doubt that such quantization will lead to much information loss. In this paper, we propose a new method to compensate for such loss. Specifically, we design hash codes based on norms of the real latent factors. 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 quantization process, we furtherly propose a model to optimize the bit of norm quantization in subsection III. Experimental results on three real world datasets, MovieLens-1M, MovieLens-10M and a subset of Netflix-100M, indicate that the recommendation accuracy of our scheme is 11%-15% higher than the best existing results.

The rest of this paper is organized as follows. In section II, we introduce some notations related to this paper and MF-L_2 recommendation model [2]. Our work is mainly based on the real latent factors learned from MF-L_2. In section III, similarity quantization and norm quantization methods will be introduced. Besides, the process of optimizing norm quantization will be described elaborately. Experimental results on three real world datasets will be shown in section IV. We choose NDCG@K as the evaluation metric of recommendation accuracy, which is widely utilized in evaluating ranking based recommendation [9]. In section V, we summarize the main contributions and the experimental results of this paper.

II. Preliminary

In this section we first introduce some notations related to recommender systems. We then introduce the most popular MF model, MF-L_2.

A. Notations

In a typical recommendation problem, we are given a users' set $U = \{u_1, u_2, \cdots, u_N\}$, an items' set $V = \{v_1, v_2, \cdots, v_M\}$ and a $M \times N$ rating matrix $\mathbf{R}$. Each element of $\mathbf{R}$, r_{ij} represents the rating of user u_i to item v_j. Generally, if user u_i has rated item v_j, r_{ij} is a positive value in a range. Otherwise, r_{ij} is equal to 0. In this paper, we assume that r_{ij} is within [0,5].

The goal of recommendation is to recommend interesting items to users according to users' past ratings to items. In this paper, we assume that rating r_{ij} represents the preference of user u_i over the item v_j. That is to say, if r_{ij} is greater than r_{ik}, then item v_j is more popular than v_k for user u_i. The aim of recommendation is searching the top-K popular items for each user. Therefore, it's necessary to predict ratings of items that are not rated by users in the past.

Note that all of the vectors in this paper represent column vectors. Uppercase bold letters express matrixes. Lowercase bold letters represent vectors. Un-bolded letters represent scalars.

B. Learning Real Latent Factors

One of the most effective personalized recommendation is based on matrix factorization (MF). In MF, users' preferences over items are modeled as inner products of real latent factors. Let the real latent factor of user u_i be represented by $\mathbf{p_i} \in \mathbb{R}^B$, the real latent factor of item v_j be expressed as $\mathbf{q_j} \in \mathbb{R}^B$. Then the preference of u_i over v_j is estimated by

$$\hat{r_{ij}} = \mathbf{p_i}^T \mathbf{q_j}. \tag{1}$$

In order to better characterize items and users by known ratings. MF models learn real latent factors by minimizing the square loss between known ratings and the estimated ratings. In order to avoid overfitting, two L_2-norm regularization terms are added in the MF objective (2) by Y. Koren, etc. [2], which was denoted by MF-L_2 model.

$$\begin{aligned}\mathbf{P}, \mathbf{Q} = &\arg\min_{\mathbf{p_i}, \mathbf{q_j}} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} (r_{ij} - \mathbf{p_i}^T \mathbf{q_j})^2 + \\ &\lambda (\sum_{i=1}^{N} \|\mathbf{p_i}\|^2 + \sum_{j=1}^{M} \|\mathbf{q_j}\|^2),\end{aligned} \tag{2}$$

where $\mathbf{P} = (\mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_N})^T$ and $\mathbf{Q} = (\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_M})$ are the learned real latent factors of users U and items V respectively. $\mathbf{R}$ is the known ratings. If user u_i has rated to item v_j in $\mathbf{R}$, $I_{ij} = 1$. Otherwise, $I_{ij} = 0$. λ is regularizer parameter.

$\mathbf{R}$ is a high sparsity matrix in general. This model utilize the known ratings $\mathbf{R}$ to predict the unknown ratings by formula (1). The above MF-L_2 model decomposes $\mathbf{R}$ into an $N \times B$ matrix $\mathbf{P}$ and a $B \times M$ matrix $\mathbf{Q}$ approximately. The aim of this model is to obtain optimal real latent factors by minimizing the objective function of (2). Due to the objective function is convex with respect to $\mathbf{P}$ and $\mathbf{Q}$ separately, $\mathbf{P}$ and $\mathbf{Q}$ can be easily obtained by coordinate descent.

III. Quantization Methods

Although MF based recommender systems can achieve high accuracy, the process of preferences ranking is time-consuming because the ranking is based on inner products of real latent factors. Hashing based recommendation can speed up the ranking process greatly. However, previous hashing recommendation was mainly based on similarity, which lead to much information loss due to ignoring the impact of norms on inner products. This section will introduce our quantization method, including similarity quantization and norm quantization. That can be regarded as the quantization of inner product. For minimizing the information loss caused by the quantization, we propose a model to learn the optimal bit for norm quantization.

A. Similarity Quantization

In section II, we have obtained the real latent factors of users and items. In this subsection we mainly focus on similarity quantization and preferences ranking problem.

Suppose the similarity based hash codes of $\mathbf{P}$ and $\mathbf{Q}$ are represented by $\mathbf{P^s}$ and $\mathbf{Q^s}$ respectively, where $\mathbf{P^s} \in \{-1,1\}^{N\times B}$ and $\mathbf{Q^s} \in \{-1,1\}^{B\times M}$. If $\mathbf{p_i}^s \in \mathbf{P^s}$ and $\mathbf{q_j}^s \in \mathbf{Q^s}$. The similarity between $\mathbf{p_i^s}$ and $\mathbf{q_j^s}$ proposed by [7] is

$$\begin{aligned} sim(\mathbf{p_i^s}, \mathbf{q_j}^s) &= \sum_{k=1}^{B} I(\mathbf{p_i^s}(k) = \mathbf{q_j^s}(k)) \\ &= \frac{1}{2}(\sum_{k=1}^{B} I(\mathbf{p_i}^s(k) = \mathbf{q_j}^s(k))) + \\ &\quad (B - (\sum_{k=1}^{B} I(\mathbf{p_i}^s(k) \neq \mathbf{q_j}^s(k)))) \\ &= \frac{1}{2}(B + \mathbf{p_i^s}^T \mathbf{q_j^s}). \end{aligned} \tag{3}$$

In the real factor space, we obtain the predicted ratings of items for each user u_i from (1),

$$\hat{r_{ij}} = \mathbf{p_i}^T \mathbf{q_j} = \|\mathbf{p_i}\| \cdot \|\mathbf{q_j}\| \cdot \cos\theta_{\mathbf{p_i},\mathbf{q_j}}.$$

We find that the predicted ratings are only associated with the cosine value and the norm of $\mathbf{q_j}$ for user u_i. On the premise that preferences can be well approximated by inner products, we recommend items based on the predicted ratings ranking. If items' ranking will be preserved when the real latent factors are transferred to binary latent factors, then the recommendation will achieve high accuracy.

As discussed above, preference based on inner product consists of two components: cosine similarity and norm of an item's real factor. Intuitively, if we can design ranking preserving hash mapping based on cosine similarity and norm separately, then the items' ranking in binary space will be consistent with that in real space. We define the ranking preserving hash mapping as follows:

Assume that a hash mapping $H : \mathbb{R} \to \{-1,1\}^Y$ which maps elements of $\mathbb{R}$ to elements of $\{-1,1\}^Y$. For $\mathbf{a},\ \mathbf{b} \in \mathbb{R}$, the corresponding images are $\mathbf{a^h},\ \mathbf{b^h} \in \{-1,1\}^Y$ respectively. We define the probe vector in $\{-1,1\}^Y$ denoted by $\mathbf{pr} = [1,1,\cdots,1]_Y{}^T$. If $\mathbf{a} \geq \mathbf{b}$, the corresponding similarities between $\mathbf{a^h},\ \mathbf{b^h},\ \mathbf{pr}$ satisfy $sim(\mathbf{a^h}, \mathbf{pr}) \geq sim(\mathbf{b^h}, \mathbf{pr})$, then H is called as ranking preserving mapping.

Based on the above consideration, for each user u_i we define H_i^s as the similarity based hash mapping. The probe vector is $\mathbf{p_i^s} = [1,1,\cdots,1]_B{}^T$, which is regarded as the similarity based hash codes of $\mathbf{p_i}$. H_i^s can be expressed as

$$\begin{aligned} &H_i^s : \mathbb{R} \to \{-1,1\}^B, \\ &\cos\theta_{\mathbf{p_i},\mathbf{q_j}} \to \mathbf{q_j^s} = [\mathbf{q_j^s}(1), \mathbf{q_j^s}(2), \cdots, \mathbf{q_j^s}(B)]^T, \end{aligned} \tag{4}$$

where

$$\mathbf{q_j^s}(m) = \begin{cases} 1\ , & if\ sgn(\mathbf{p_i}(m)) = sgn(\mathbf{q_j}(m)), \\ -1\ , & else, \end{cases}$$

and $m \in \{1,2,\cdots,B\}$.

Through the above similarity based hash mapping, we can get similarity based hash codes of items and users with low time complexity. Because we only need to compare the sign of real latent factors. Therefore, such hash mapping can speed up recommendation. Furthermore, the probe vector will search the most similar vectors to $\mathbf{p_i}$. Because the probe vector $\mathbf{p_i^s}$ can be regarded as an evaluation metric of similarities between the hash codes of users and items. If $sim(\mathbf{p_i^s}, \mathbf{q_j^s}) \geq sim(\mathbf{p_i^s}, \mathbf{q_k^s})$, then $\mathbf{q_j^s}$ is more similar to $\mathbf{p_i^s}$ compared with $\mathbf{q_k^s}$. It can also be regarded as an evaluation metric of cosine similarity in real latent space. If $sim(\mathbf{p_i^s}, \mathbf{q_j^s}) \geq sim(\mathbf{p_i^s}, \mathbf{q_k^s})$, then $\mathbf{q_j}$ has more same sign elements with $\mathbf{p_i}$ compared to $\mathbf{q_k}$. Therefore, the angle between $\mathbf{p_i}$ and $\mathbf{q_j}$ is smaller than the angle between $\mathbf{p_i}$ and $\mathbf{q_k}$. We can further conclude $\cos\theta_{\mathbf{p_i},\mathbf{q_j}} \geq \cos\theta_{\mathbf{p_i},\mathbf{q_k}}$, and vice versa.

It can be validated that H_i^s is ranking preserving. To keep the paper reasonably concise, the validation process is not mentioned in this paper.

B. Norm Quantization

In this subsection we will propose a new norm quantization method to generate hash codes from norms of real latent factors, which increases recommendation accuracy to a great extent.

As discussed in the above subsection, in order to preserve items' ranking, we need to design ranking preserving hash mapping based on norms of real latent factors. Therefore, we define H_i^n as the norm based hash mapping for each user u_i. The probe vector is $\mathbf{p_i^n} = [1,1,\cdots,1]_K{}^T$, which can be also looked as the norm based hash codes for $\mathbf{p_i}$. H_i^n is expressed as

$$\begin{aligned} &H_i^n : \mathbb{R} \to \{-1,1\}^K, \\ &\|\mathbf{q_j}\| \to \mathbf{q_j^n} = [\mathbf{q_j^n}(1), \mathbf{q_j^n}(2), \cdots, \mathbf{q_j^n}(K)]^T, \end{aligned} \tag{5}$$

where

$$\mathbf{q_j^n}(t) = \begin{cases} 1 \quad , & if \, \|\mathbf{q_j}\| \geq \dfrac{10t-5}{2K}, \\ -1, & else, \end{cases} \tag{6}$$

$t \in \{1, 2, \cdots, K\}$, K is the bit of norm quantization. The value of K will be determined in the following subsection. Our method is designed under an assumption that all norms are within [0,5]. It's an appropriate assumption because the norms of $\mathbf{p_i}$ and $\mathbf{q_j}$ are restricted by L_2-norm regularization terms in (2). Therefore, most of the learned latent factors, $\mathbf{p_i}$ and $\mathbf{q_j}$ have small norms. Besides, for norms with more than 5, that's also effective. Norms with greater than 5 will be transferred to $[1, 1, \cdots, 1]_K^{\ T}$ by our quantization method.

Through the above norm based hash mapping, we can obtain $K-$bit norm based hash codes that can achieve higher accuracy than 2-bit norm based codes in PPH, because K-bit quantization is more refined. Besides, the probe vector $\mathbf{p_i^n}$ can be used to search the most similar items to user u_i.

We can validate that the above norm based hash mapping is ranking preserving. For each user u_i and any two items v_j and v_k, the corresponding real latent factors are $\mathbf{p_i}$, $\mathbf{q_j}$ and $\mathbf{q_k}$ respectively. If $\|\mathbf{q_j}\| \geq \|\mathbf{q_k}\|$. From (5) and (6), we can attain that $\left\|\mathbf{q_j^n}\right\|$ has more bits with 1 than $\|\mathbf{q_k^n}\|$. Therefore,

$$sim(\mathbf{p_i^n}, \mathbf{q_j^n}) \geq sim(\mathbf{p_i^n}, \mathbf{q_k^n}).$$

Hence, the norm based mapping is ranking preserving.

For example, for a particular user u_i and two items v_j and v_k, let the corresponding real latent factors be $\mathbf{p_i}$, $\mathbf{q_j}$ and $\mathbf{q_k}$ respectively, $\|\mathbf{p_i}\| = 2.1$, $\|\mathbf{q_j}\| = 3.2$ and $\|\mathbf{q_k}\| = 3.3$. Suppose $K = 10$ (which will be determined in the next subsection), then we can calculate the norm based hash codes listed below by (5) and (6),

$$\begin{aligned} \mathbf{p_i}^n &= [1, \ 1, \ 1, \ 1, \ 1, \ 1, \ 1, \ 1, \ 1, \ 1], \\ \mathbf{q_j}^n &= [1, \ 1, \ 1, \ 1, \ 1, \ 1, -1, -1, -1, -1], \\ \mathbf{q_k}^n &= [1, \ 1, \ 1, \ 1, \ 1, \ 1, \ 1, -1, -1, -1]. \end{aligned}$$

The norms $\|\mathbf{q_j}\| \leq \|\mathbf{q_k}\|$, the similarity between the corresponding hash codes satisfy

$$sim(\mathbf{p_i}^n, \mathbf{q_j}^n) \leq sim(\mathbf{p_i}^n, \mathbf{q_k}^n).$$

Therefore, the above norm based hashing mapping is ranking preserving.

C. Learning the Optimal Bit

The similarity based hash codes usually lead to much accuracy loss. In order to offset the loss, we discussed the norm quantization method in the above subsection. In this subsection, we will discuss how many bits of norm quantization are appropriate for given dimension B. As a solution, we propose a model to learn the optimal bit by minimizing the accuracy loss. To our knowledge, this is the first time to learn the optimal bit for norm quantization. Substituted the learned optimal bit K into norm quantization steps in the above subsection. The loss of recommendation accuracy can be reduced significantly.

As discussed in the above subsection, similarity based hashing mapping and norm based hashing mapping are ranking preserving, separately. Based on the linear approximation assumption mentioned by PPH [3], hash codes in this paper are also composed of two parts: the norm based hash codes and the similarity based hash codes. Assume that hash codes for each $\mathbf{p_i}$ and $\mathbf{q_j}$ can be expressed as

$$\mathbf{p_i^h} = [\mathbf{p_i^n}, \mathbf{p_i^s}], \quad \mathbf{q_j^h} = [\mathbf{q_j^n}, \mathbf{q_j^s}],$$

where each $\mathbf{p_i^n}, \mathbf{p_i^s}, \mathbf{q_j^n}, \mathbf{q_j^s}$ can be obtained from (4), (5) and (6). If the total length of hash codes is L, from formula (3) the similarity between $\mathbf{p_i^h} \in \{-1, 1\}^L$ and $\mathbf{q_j^h} \in \{-1, 1\}^L$ can be expressed as

$$sim(\mathbf{p_i^h}, \mathbf{q_j^h}) = \frac{1}{2}(L + \mathbf{p_i^h}^{\ T}\mathbf{q_j^h}). \tag{7}$$

Assume that each norm based hash code has K bits and each similarity based hash codes has B bits. From (7), the similarity between $\mathbf{p_i^h}$ and $\mathbf{q_j^h}$ can be rewritten as

$$sim(\mathbf{p_i^h}, \mathbf{q_j^h}) = \frac{1}{2}(K + B + \mathbf{p_i^h}^{\ T}\mathbf{q_j^h}). \tag{8}$$

In the real latent space, the predicted ratings are in the range of $[0, 5]$ (ratings in our experiments of this paper are within $[0, 5]$). While in binary latent space, the preferences are approximated by similarities in (8) that take values in $[0, K + B]$. For Data Consistency, we need to normalize the inner products and the similarities by deviation method. In order to minimize the information loss caused by hashing quantization, we obtain the following loss function,

$$Loss(K) = \left(\frac{1}{5}\mathbf{p_i}^T\mathbf{q_j} - \frac{1}{2} - \frac{1}{2(K+B)}\mathbf{p_i^h}^{\ T}\mathbf{q_j^h}\right)^2.$$

We can get the optimal K by minimizing the above loss function via iterative approach. But it will lead poor generalization due to overfitting. In order to avoid overfitting, we add a regularization term to penalize large K. Besides, the constraint of K can also reduce the complexity of norm quantization. Therefore, we get the following objective.

$$L(K) = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\frac{1}{5}\mathbf{p_i}^T\mathbf{q_j} - \frac{1}{2} - \frac{1}{2(K+B)}\mathbf{p_i^h}^T\mathbf{q_j^h}\right)^2 + \beta K, \tag{9}$$

By minimizing the above function, we can get the optimal bit K for given B,

$$K = \arg\min_K \left(L(K)\right).$$

where β is a constant selected by cross-validation which weighs generalization and overfitting. In this paper, we set $\beta = 7$. The above problem can be solved by Algorithm 1, where K_{max} is the times of iterations, we set $K_{max} = 100$ because bigger K will lead to higher quantization complexity.

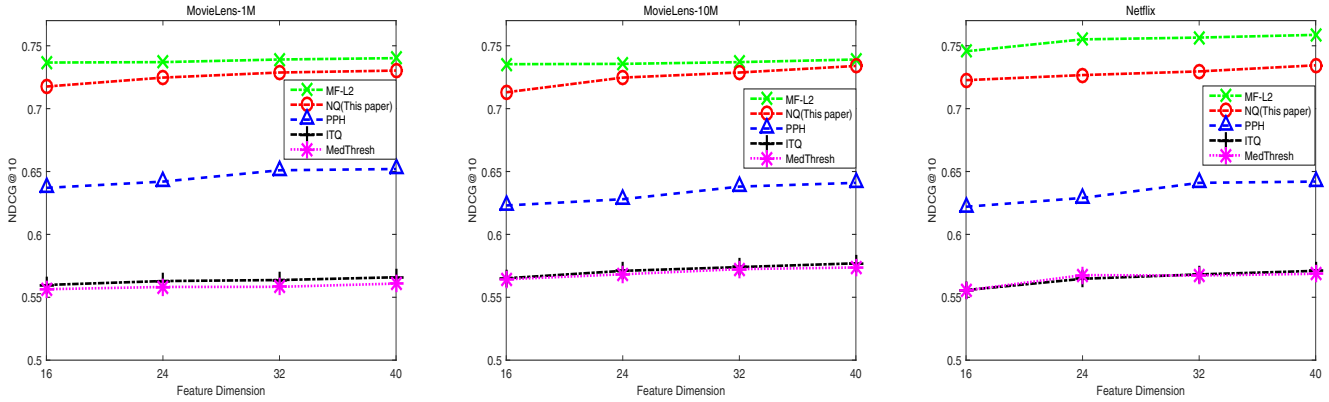

Fig. 1. Recommendation performance(NDCG@10) on MovieLens-1M, 10M, a subset of Netflix dataset

Algorithm 1 Optimizing Norm Quantization

1: Input:B, $\mathbf{P}$, $\mathbf{Q}$, β. (B, $\mathbf{P}$, $\mathbf{Q}$ can be obtained from (2)).
2: **for** $K = 1$ to K_{max}
3: Compute the objective $L(K)$.
4: **end**
5: $K = \underset{K}{\text{argmin}}\ (L(K))$
6: Output K.

IV. Experiments

In this section, we mainly introduce our experiment settings and analyze some results. Experiments on three real world datasets show that our quantization scheme can achieve much higher recommendation accuracy than previous quantization methods.

A. Datasets and Evaluation Metric

We evaluate our method on three real world open datasets: MovieLens-1M, MovieLens-10M, and a subset of Netflix-100M. 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 of these ratings are within [0,5]. Netflix-100M dataset is one of the biggest recommendation open dataset that contains 100,480,507 ratings from 480,189 users to 17770 thousand movies.

We choose NDCG@K [10] to evaluate the recommendation accuracy, which has been widely used to evaluate the quantity of predicted ranking. As a matter of fact, recommendation can also be regarded as items' ranking for each user. In our experiments, we predicted the top-10 popular items for each user. Therefore, we evaluate the recommendation accuracy by NDCG@10.

B. Experiment Settings

In our experiments, we divide each of the above three datasets into 3 disjoint subsets randomly: training dataset, validation dataset and testing dataset. Training datasets are composed of 20 ratings of each user. Validation datasets include 10 ratings of each user. The rest at least 10 ratings of each user make up the testing datasets, because we need to predict the top-10 popular items for each user. Thus, users having no less than 40 ratings are considered in our experiments.

Training datasets and validation datasets are utilized to learn the real latent factors by model (2). Validation datasets are used to select the regularization parameter λ. Experiments show that the recommendation accuracy on validation datasets is quite stable within a large range of λ round the optimal one. In this paper, we choose λ=12 for all datasets. Testing datasets are used to test the recommendation accuracy of our method under the evaluation NDCG@10.

In the process of learning the optimal bit K for norm quantization, we divide the learned real latent factors $\mathbf{P}$ and $\mathbf{Q}$ into 3 disjoint datasets respectively: training dataset, validation dataset and testing dataset. Similarity, we choose β by cross-validation. Experiments show that β varies with B.

C. Experimental Results

Figure 1 shows that the performances (NDCG@10) of NQ (method in this paper) on three datasets are superior to the previous three hashing methods. MedThresh and ITQ were implemented by K. Zhou [7], [8] and PPH was proposed by Zhang et al. [3]. Moreover, we conclude that the recommendation accuracy of NQ is very close to MF-L_2 method. We make the following observations from the results:

(1) Compared with other hash quantization methods, our method has much higher recommendation accuracy than them. Other hash codes are mainly generated from similarity quantization. However, similarities between real latent factors of users and items can not be well approximated as preferences. The proposed model outperforms previous methods mainly because it quantizes both norms and cosine similarity in real latent space. The norm quantization process can compensate the accuracy loss caused by similarity quantization. Specifically, the process of learning the optimal bit can minimize the information loss for given B.

(2) Compared with MF-L_2 method, the accuracy of our method is very close to the classic MF-L_2 method. Due to

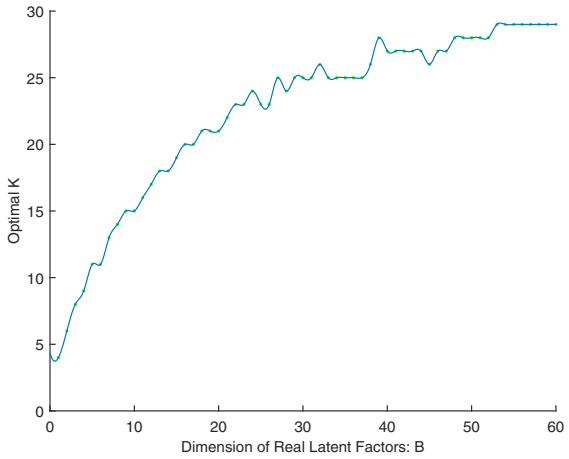

Fig. 2. Optimal K varies with B on MovieLens-1M

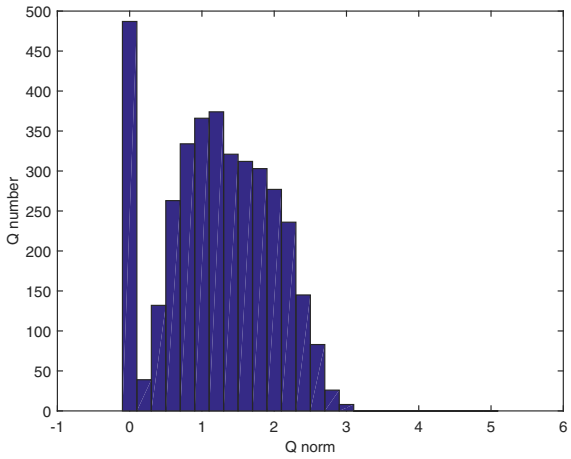

Fig. 3. The norm distribution of vectors in $\mathbf{Q}$ with $B = 24$ on MovieLens-1M

fast search speed, hash technologies are more popular than MF-L_2 in recommender systems.

Based on Algorithm 1, we can get the optimal bits K for different dimensions B. In Figure 2 we show the optimal K varies with the dimension B on MovieLens-1M. We can conclude:

(1) As B is small, similarity based hash codes lead to much information loss. As a compensation for the loss, the optimal K increases faster than B. Norm quantization will offsets the loss significantly.

(2) As B rises up, information loss caused by similarity quantization is reduced, meanwhile, the complexity of quantization is increased. Therefore, the optimal K increases slower than B.

(3) As $B \geq 40$, smaller information loss and higher complexity of norm quantization impel the optimal K to be stabilized gradually. Similarity quantization plays dominant role in recommendation.

Through learning the optimal K for various B, the information loss caused by similarity quantization will be compensated by norm quantization automatically. Therefore, our hashing based recommendation can achieve higher accuracy than previous hashing approaches.

Based on MF-L_2, we can obtain the real latent factors $\mathbf{P}$ and $\mathbf{Q}$ by coordinate decent. In order to verify the rationality of norms' assumption in section III, we show the norms' distribution of $\mathbf{Q}$ in Figure 3. We can find that almost all norms are within $[0, 5]$. Therefore, our assumption is reasonable.

V. Conclusion

In this paper, we proposed a new binary quantization method that consists of similarity quantization and norm quantization. The norm quantization can compensate for similarity quantization significantly. Furthermore, we develop a model to learn the optimal bit for norm quantization that can minimize the information loss caused by the entire hash codes. Experimental results show that our hashing recommendation method is superior to other hashing methods on three real world datasets. In addition, we get a rule from a large number of experiments on learning the optimal bit of norm quantization. When the dimension of real latent factors is small, the norm quantization is important because it can offset the information loss caused by similarity quantization. Otherwise, when the dimension of real features rises up, similarity quantization plays dominant role in recommendation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61272175, No. 61572109) and the 863 High Technology Plan (Grant No. 2015AA01A707).

References

[1] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[3] Z. Zhang, Q. Wang, L. Ruan, and L. Si, "Preference preserving hashing for efficient recommendation," in *Proc. ACM SIGIR*, 2014, pp. 183–192.

[4] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Content-aware collaborative filtering for location recommendation based on human mobility data," in *Proc. IEEE ICDM*, Nov. 2015, pp. 261–270.

[5] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *arXiv preprint arXiv: 1408.2927*, 2014.

[6] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *Proc. ACM WWW*, 2007, pp. 271–280.

[7] K. Zhou and H. Zha, "Learning binary codes for collaborative filtering," in *Proc. ACM SIGKDD*, 2012, pp. 498–506.

[8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[9] M. Volkovs and R. S. Zemel, "Collaborative ranking with 17 parameters," in *Proc. NIPS*, 2012, pp. 2294–2302.

[10] G.-J. Qi, X.-S. Hua, and H.-J. Zhang, "Learning semantic distance from community-tagged media collection," in *Proc. ACM MM*, 2009, pp. 243–252.