

# COLLABORATIVE GENERATED HASHING FOR MARKET ANALYSIS AND FAST COLD-START RECOMMENDATION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Cold-start and efficiency issues of the Top-k recommendation are critical to large-scale recommender systems. Previous hybrid recommendation methods are effective to deal with the cold-start issues by extracting real latent factors of cold-start items(users) from side information, but they still suffer low efficiency in online recommendation caused by the expensive similarity search in real latent space. This paper presents a collaborative generated hashing (CGH) to improve the efficiency by denoting users and items as binary codes, which applies to various settings: cold-start users, cold-start items and warm-start ones. Specifically, CGH is designed to learn hash functions of users and items through the Minimum Description Length (MDL) principle; thus, it can deal with various recommendation settings. In addition, CGH initiates a new marketing strategy through mining potential users by a generative step. To reconstruct effective users, the MDL principle is used to learn compact and informative binary codes from the content data. Extensive experiments on two public datasets show the advantages for recommendations in various settings over competing baselines and analyze the feasibility of the application in marketing.

## 1 INTRODUCTION

With the explosion of e-commerce, most customers are accustomed to receiving a variety of recommendations, such as movies, books, news, or hotels they might be interested in. Traditional recommender systems just recommended items that are similar to what they liked or rated in the previous. Recommendations help users find their desirable items, and also creates new revenue opportunities for vendors, such as Amazon, Taobao, eBay, etc. Among them, one of the most popular recommendation methods, collaborative filtering is dependent on a large amount of user-item interactive information to provide an accurate recommendation. However, most of new e-commerce vendors do not have enough interactive data, which leads to low recommendation accuracy, i.e., cold-start issues.

Previous studies on cold-start issues generally modeled as a combination of collaborative filtering and content filtering, known as hybrid recommender systems. Specifically, they learned real latent factors by incorporating the side information into the interactive data. Such as Collaborative Deep Learning (CDL) (Wang et al., 2015), Visual Bayesian Personalized Ranking (VBPR) (He & McAuley, 2016), Collaborative Topic modeling for Recommendation (CTR) (Wang & Blei, 2011), and the DropoutNet for addressing cold start (DropoutNet)(Volkovs et al., 2017), ABCPRec for Bridging Consumer and Producer Roles for User-Generated Content Recommendation (ABCPRec)(Tsukuda et al., 2019). All of the above hybrid recommender systems were modeled in real latent space, which leads to low efficiency for the online recommendation with the increasing scale of datasets.

Recent studies show the promising of hashing based methods to tackle the efficiency challenge by representing users and items with binary codes (Zhang et al., 2014; Zhou & Zha, 2012; Zhang et al., 2016; Liu et al., 2019), because the preference score can be represented by the Hamming distance calculated via XOR operation efficient (Wang et al., 2014). However, the existing hashing based recommendations are learning-based frameworks, which leads to NP-hard problems of optimizing

discrete objectives. Thus many scholars learned binary codes by some approximate techniques, such as the two-stage hashing learning method utilized in Preference Preserving Hashing(PPH) (Zhang et al., 2014) and the Iterative Quantization(ITQ) (Zhou & Zha, 2012). To reduce information loss, two learning-based hashing frameworks: bit-wise learning and block-wise learning were respectively proposed in hashing based recommendation frameworks (Zhang et al., 2016; Wang et al., 2019; Zhang et al., 2018; Zheng et al.).

However, due to the requirement of binary outputs for learning-based hashing frameworks, the training procedure is expensive for large-scale recommendation, which motivates us to propose a generative approach to learn hash functions. In this paper, we propose the collaborative generated hashing(CGH) to learn hash functions of users and items from content data with the principle of Minimum Description Length (MDL) (Dai et al., 2017).

In marketing area, mining potential customers is crucial to the e-commerce. CGH provides a strategy to discover potential users by the generative step. To reconstruct effective users, uncorrelated and balanced limits are imposed to learn compact and informative binary codes with the principle of the MDL. Especially, discovering potential customers is vital to the success of adding new items for a recommendation platform (Papies et al., 2017). Specifically, for a new item, we can generate a new potential user by the generative step (detailed in Section 2.1), and then search the nearest potential users in the user set. By recommending a new product to the potential users who might be interested in but didn't plan to buy, further e-commerce strategies can be developed to attract those potential users.

We organize the paper as follows: Section 2 introduce the main techniques of CGH. We first introduce the framework of CGH and compare it with the closely related competing baselines: CDL (Wang et al., 2015) and DropoutNet (Volkovs et al., 2017); we then formulate the generative step in Section 2.1 and the inference step in Section 2.2, respectively; we finally summarize the training objective and introduce the optimization in Section 2.3. Particularly, we demonstrate the process of mining potential users for the marketing application in Section 2.1. Section 3 presents the experimental results for marketing analysis and recommendation accuracy in various settings. Section 4 concludes the paper.

The main contributions of this paper are summarized as follows:

- (1) We propose the Collaborative Generated Hashing (CGH) with the principle of MDL to learn compact but informative hash codes, which applies to various settings for recommendation.
- (2) We provides a marketing strategy by discovering potential users by the generative step of CGH, which can be applied to boost the e-commerce development.
- (3) We evaluate the effectiveness of the proposed CGH compared with the state-of-the-art baselines, and demonstrate its robustness and convergence properties on the public datasets.

## 2 COLLABORATIVE GENERATED HASHING

The framework of the proposed CGH is shown in Fig. 1(c), where  $U$ ,  $V$  and  $R$  are respectively observed user content, item content and rating matrix.  $B$  and  $D$  are binary codes of users and items, respectively. CGH consists of the generative step marked as dashed lines and the inference step denoted by solid lines. Once training is finished, we fix the model and make forward passes to obtain binary codes  $B$  and  $D$  through the inference step, and then conduct recommendation. For the marketing application, we create a new user via the generative step.

In comparison of CGH with the closely related baseline CDL (Wang et al., 2015), the proposed CGH aims to learn binary codes instead of real latent vectors  $P$  and  $Q$  due to the advantage of hashing for online recommendation; plus the CGH optimizes an objective with the principle of MDL, while CDL optimized the joint objective of rating loss and item content reconstruction error. In comparison of CGH with DropoutNet (Volkovs et al., 2017), CGH can be used as a marketing strategy by discovering potential users; plus CGH learns hash functions by stacked denoising autoencoder, while DropoutNet obtained real latent factors by the standard neural network.

In the following we start by first formulating the generative process and demonstrating the application in marketing area; we then formulate the inference step; we finally summarize the training objective and the optimization method.

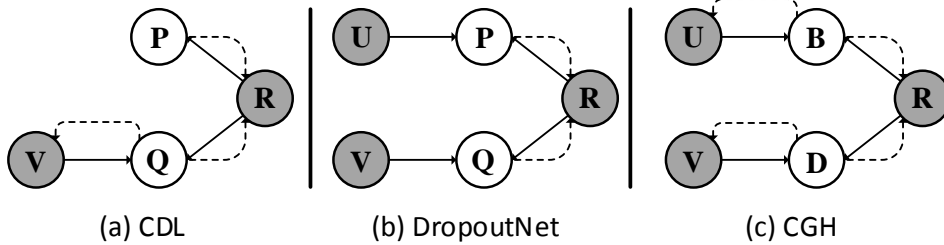


Figure 1: Differences between CDL, DropoutNet and our proposed CGH. Solid lines and dashed lines respectively represent the inference (encoding) and the generative (decoding) process. The shaded nodes  $U, V, R$  are observed user content, item content and rating, respectively.  $P(B), Q(D)$  denotes real latent factors(binary codes) of users and items.

### 2.1 MINING POTENTIAL USERS

. Give a sparse rating matrix  $R$  and item content data  $V \in \mathbb{R}^{d_v}$ , where  $d_v$  is the dimension of the content vector, and  $V$  is stacked by the bag-of-words vectors of item content in the item set  $V$ . most previous studies were focus on modeling deterministic frameworks to learn representations of items for item recommendation, such as CDL, CTR, DropoutNet, et.al. In this paper, we discover a new strategy from a perspective of marketing for item recommendation – mining potential users.

We demonstrate the process of mining potential users by an item through the generative step in Fig. 2. After the inference step, the binary code of item  $j$  is available. By maximizing the similarity function  $\delta(\mathbf{b}_i, \mathbf{d}_j)$  (detailed in Section 2.1), the optimal binary code  $\mathbf{b}_p$  is obtained. Then we generate the new user  $\mathbf{u}_p$  through the generative step. Finally we find out potential users from the user set by some nearest neighborhood algorithms, such as KNN. As a marketing strategy, it can discover potential users for both warm-start items and cold-start items. Thus, from a perspective of marketing, it can be regarded as another kind of item recommendation.

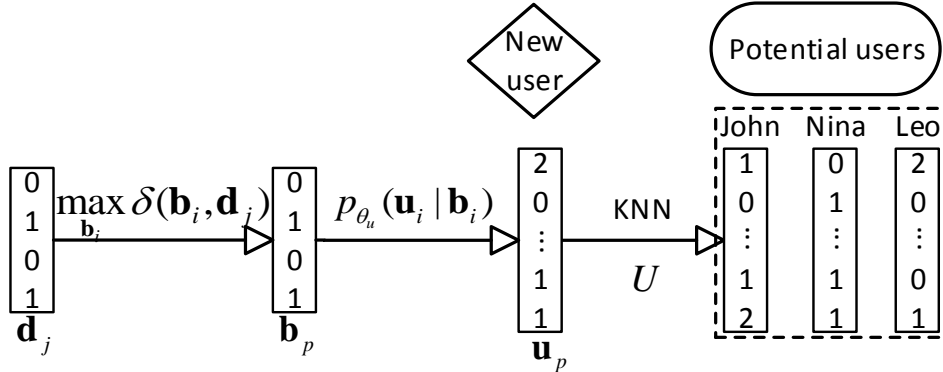


Figure 2: Demonstration of mining potential users for an item  $j$ . After the inference step,  $\mathbf{d}_j$  is available, we first find out the most similar binary code  $\mathbf{b}_p$ ; we then generate a new potential user  $\mathbf{u}_p$  by the generative process; we furtherly search the top- $k$  nearest potential users from the user set by some nearest neighborhood algorithms (e.g., KNN).

The generation process (also referred as decoding process) is denoted by dashed lines in Fig. 1 (c). Fix binary codes  $\mathbf{b}_i$  and  $\mathbf{d}_j$  of the user  $i$  and the item  $j$ , the bag-of-words vector  $\mathbf{u}_i$  of the user  $i$  ( $\mathbf{v}_j$  of the item  $j$ ) is generated via  $p(\theta_u)$ . The ratings  $r_{ij}$  is generated by  $\mathbf{b}_i$  and  $\mathbf{d}_j$ . We use a simple Gaussian distribution to model the generation of  $\mathbf{u}_i$  and  $\mathbf{v}_j$  given  $\mathbf{b}_i$  and  $\mathbf{d}_j$  like Stochastic Generative Hashing (SGH) (Dai et al., 2017):

$$\begin{aligned}
 p(\mathbf{u}_i|\mathbf{b}_i) &= \mathcal{N}(\mathcal{C}_u \mathbf{b}_i, \lambda_u^{-1} \mathbf{I}), \\
 p(\mathbf{v}_j|\mathbf{d}_j) &= \mathcal{N}(\mathcal{C}_v \mathbf{d}_j, \lambda_v^{-1} \mathbf{I}),
 \end{aligned}
 \tag{1}$$

where  $\mathcal{C}_{uk} = [\mathbf{c}_{uk}]_{k=1}^r$ ,  $\mathbf{c}_{uk} \in \mathbb{R}^{d_u}$  is the codebook (Dai et al., 2017) with  $r$  codewords, which is similar for  $\mathcal{C}_v$ , and  $d_u$  is the dimension of the bag-of-the-words vector of users. The prior is modeled as the multivariate Bernoulli distribution on hash codes:  $p(\mathbf{b}_i) \sim \mathcal{B}(\rho_u)$ , and  $p(\mathbf{d}_j) \sim \mathcal{B}(\rho_v)$ , thus the prior probability are as follows:

$$\begin{aligned} p(\mathbf{b}_i) &= \prod_{k=1}^r \rho_{uik}^{b_{ik}} (1 - \rho_{uik})^{1-b_{ik}}, \\ p(\mathbf{d}_j) &= \prod_{k=1}^r \rho_{vj}^{d_{jk}} (1 - \rho_{vj})^{1-d_{jk}}, \end{aligned} \quad (2)$$

We formulate the rating with the similarity between binary codes of users and items like the most successful recommender systems, matrix factorization (Koren et al., 2009). Then the rating is thus drawn from the normal distribution centered at the similarity value,

$$p(r_{ij} | \mathbf{b}_i, \mathbf{d}_j) \sim \mathcal{N}(\delta(\mathbf{b}_i, \mathbf{d}_j), C_{ij}^{-1}), \quad (3)$$

where  $\delta(\mathbf{b}_i, \mathbf{d}_j) = 1 - \frac{1}{r} \text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$  denotes the similarity between binary codes  $\mathbf{b}_i$  and  $\mathbf{d}_j$ .  $\text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$  represents Hamming distance between the two binary vectors, which has been widely applied in hashing-based recommender system (Wang & Blei, 2011; Lian et al., 2017; Zhang et al., 2018).  $C_{ij}$  is the precision parameter that serves as confidence for  $r_{ij}$  similar to that in CTR (Wang & Blei, 2011) ( $C_{ij} = a$  if  $r_{ij} = 1$  and  $C_{ij} = b$  otherwise) due to the fact that  $r_{ij} = 0$  means the user  $i$  is either not interested in item  $j$  or not aware of it.

With the generative model constructed, the joint probability of both observed ratings, content vectors and binary codes is given by

$$p(\mathbf{R}, \mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{D}) = \prod_{i,j} p(r_{ij} | \mathbf{b}_i, \mathbf{d}_j) p(\mathbf{u}_i | \mathbf{b}_i) p(\mathbf{v}_j | \mathbf{d}_j) p(\mathbf{b}_i) p(\mathbf{d}_j) \quad (4)$$

## 2.2 CONSTRAINTS ON BINARY LATENT VARIABLES

The inference process (also referred as encoding process) shown in the Fig. 1 (c) with dashed lines, the binary latent variables  $\mathbf{b}_i$  ( $\mathbf{d}_j$ ) depends on the content vector  $\mathbf{u}_i$  ( $\mathbf{v}_j$ ) and the rating  $\mathbf{R}$  (shadowed in Fig 1). Inspired by the recent work on generative hashing (Dai et al., 2017) and DropoutNet (Volkovs et al., 2017), we use a multivariate Bernoulli distribution to model the inference process of  $\mathbf{b}_i$  and  $\mathbf{d}_j$  with linear parametrization, i.e.,

$$\begin{aligned} q(\mathbf{b}_i | \tilde{\mathbf{u}}_i) &= \mathcal{B}(\sigma(\mathcal{T}_u^T \tilde{\mathbf{u}}_i)) \\ q(\mathbf{d}_j | \tilde{\mathbf{v}}_j) &= \mathcal{B}(\sigma(\mathcal{T}_v^T \tilde{\mathbf{v}}_j)), \end{aligned} \quad (5)$$

where  $\tilde{\mathbf{u}}_i = [\mathbf{u}_i, \mathbf{p}_i]$ ,  $\tilde{\mathbf{v}}_j = [\mathbf{v}_j, \mathbf{q}_j]$ .  $\mathbf{p}_i$  and  $\mathbf{q}_j$  are the results of  $r$ -dimension matrix factorization (Koren et al., 2009) of  $\mathbf{R}$ , i.e.,  $r_{ij} \approx \mathbf{p}_i^T \mathbf{q}_j$ .  $\mathcal{T}_u = [\mathbf{t}_{uk}]_{k=1}^r$ ,  $\mathbf{t}_{uk} \in \mathbb{R}^{d_u+r}$ ,  $\mathcal{T}_v = [\mathbf{t}_{vk}]_{k=1}^r$ ,  $\mathbf{t}_{vk} \in \mathbb{R}^{d_v+r}$  are the transformation matrices of linear parametrization. From SGH (Dai et al., 2017), the MAP solution of the eq. (5) is readily given by

$$\begin{aligned} \mathbf{b}_i &= \underset{\mathbf{b}_i}{\operatorname{argmax}} q(\mathbf{b}_i | \mathbf{u}_i) = \frac{\operatorname{sign}(\mathcal{T}_u^T \mathbf{u}_i) + 1}{2}, \\ \mathbf{d}_j &= \underset{\mathbf{d}_j}{\operatorname{argmax}} q(\mathbf{d}_j | \mathbf{v}_j) = \frac{\operatorname{sign}(\mathcal{T}_v^T \mathbf{v}_j) + 1}{2}. \end{aligned} \quad (6)$$

With the linear projection followed by a sign function, we can easily get hash codes of users and items. However, hashing with a simple sign function suffers from large information loss according to (Zhang et al., 2016), which motivates us to add constraints on parameters in the inference step.

To derive compact and informative hash codes for users and items, we add balanced and uncorrelated constraints in the inference step. The balanced constraint is proposed to maximize the entropy of each binary bit (Zhou & Zha, 2012), and the uncorrelated constraint makes each bit is independent of others. Then we can obtain compact and informative hash codes by the following constraints,

$$\begin{aligned} \text{Balanced constraint: } & \sum_k b_{ik} = 0, \sum_k d_{jk} = 0 \\ \text{Uncorrelated constraint: } & \mathbf{b}_i \mathbf{b}_i^T = \mathbf{I}_r, \mathbf{d}_j \mathbf{d}_j^T = \mathbf{I}_r \end{aligned} \quad (7)$$

From the eq. (6),  $\mathbf{b}_i$  and  $\mathbf{d}_j$  are only dependent on parameters  $\mathcal{T}_u$  and  $\mathcal{T}_v$ , respectively, thus we add constraints on  $\mathcal{T}_u$  and  $\mathcal{T}_v$  directly. So eq. (7) is equivalent to the following constraints,

$$\begin{aligned} \text{Balanced constraint: } & \mathcal{T}_u^T \mathbf{1}_r = \mathbf{0}, \mathcal{T}_v^T \mathbf{1}_r = \mathbf{0}, \\ \text{Uncorrelated constraint: } & \mathcal{T}_u^T \mathcal{T}_u = \mathbf{I}_{d_u+r}, \mathcal{T}_v^T \mathcal{T}_v = \mathbf{I}_{d_v+r}. \end{aligned} \quad (8)$$

By imposing the above constraints in the training step, compact and informative hash codes can be obtained through the inference process. Next we summarize the training objective and its optimization.

### 2.3 TRAINING OF CGH

Since our goal is to reconstruct users, items and ratings by using the least information of binary codes, we train the CGH with the MDL principle, which finds the best parameters that maximally compress the training data and meanwhile keep the information carried, thus CGH aims to minimize the expected amount of informations related to  $q$ :

$$\begin{aligned} \mathcal{L}(q) &= E_q[\log p(\mathbf{R}, \mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{D}) - \log q(\mathbf{B}, \mathbf{D})] \\ &= E_q[\log p(\mathbf{R}|\mathbf{B}, \mathbf{D}) + \log p(\mathbf{U}|\mathbf{B}) + \log p(\mathbf{V}|\mathbf{D}) - KL(q(\mathbf{B}|\tilde{\mathbf{U}})||p(\mathbf{B})) - \\ & \quad KL(q(\mathbf{D}|\tilde{\mathbf{V}})||p(\mathbf{D}))] \end{aligned} \quad (9)$$

Maximizing the posterior probability is equivalent to maximizing  $\mathcal{L}(q)$  by only considering the variational distribution of  $q(\mathbf{B}, \mathbf{D})$ , the objective becomes

$$\begin{aligned} \mathcal{L}_{MAP}(\Theta, \Phi) &= - \sum_{i,j} \frac{C_{ij}}{2} (r_{ij} - \delta(\mathbf{b}_i, \mathbf{d}_j))^2 - \frac{\lambda_u}{2} \sum_i (u_i - C_u \mathbf{b}_i)^2 - \frac{\lambda_v}{2} \sum_j (v_j - C_v \mathbf{d}_j)^2 \\ & \quad - KL(q_{\phi_u}||p_{\theta_u}) - KL(q_{\phi_v}||p_{\theta_v}) - \nabla(\Theta, \Phi) \end{aligned} \quad (10)$$

where  $\Theta = \{\theta_u, \theta_v\}$ ,  $\Phi = \{\phi_u, \phi_v\}$ ,  $\nabla(\Theta, \Phi)$  is the regularizer term with parameters  $\Theta$  and  $\Phi$ . By training the objective in eq. (10), we obtain binary codes, but some bits probably be correlated. To minimize the reconstruction error, SGH had to set up the code length as long as  $r = 200$ . Our goal in this paper is to obtain compact and informative hash codes, thus we impose the balance and independent constraints on hash codes by eq. (8). Maximizing the eq. (10) is transformed to minimizing the following constrained objective function of the proposed Collaborative Generative Hashing (CGH),

$$\begin{aligned} \mathcal{L}_{CGH}(\Theta, \Phi) &= \sum_{i,j} \frac{C_{ij}}{2} (r_{ij} - \delta(\mathbf{b}_i, \mathbf{d}_j))^2 + \frac{\lambda_u}{2} \sum_i (u_i - C_u \mathbf{b}_i)^2 + \frac{\lambda_v}{2} \sum_j (v_j - C_v \mathbf{d}_j)^2 + \\ & \quad KL(q_{\phi_u}||p_{\theta_u}) + KL(q_{\phi_v}||p_{\theta_v}) + \alpha_u \|\mathcal{T}_u^T \mathbf{1}_r\|_2^2 + \alpha_v \|\mathcal{T}_v^T \mathbf{1}_r\|_2^2 + \\ & \quad \beta_u \|\mathcal{T}_u^T \mathcal{T}_u - \mathbf{I}_{d_u+r}\|_2^2 + \beta_v \|\mathcal{T}_v^T \mathcal{T}_v - \mathbf{I}_{d_v+r}\|_2^2 + \nabla(\Theta, \Phi). \end{aligned} \quad (11)$$

The objective of CGH in eq. (11) is a discrete optimization problem, which is difficult to optimize straightforwardly, so in the training stage, the tanh function is utilized to replace the sign function in the inference step, and then the continuous outputs are used as a relaxation of hash codes.

With the relaxation, we train all components jointly with back-propagation. After training, we fix them and make forward passes to map the concatenate vectors in  $\tilde{\mathbf{U}}$  and  $\tilde{\mathbf{V}}$  to binary codes  $\mathbf{B}$  and  $\mathbf{D}$ , respectively. The recommendation in various settings is then conducted using  $\mathbf{B}$  and  $\mathbf{D}$  by the similarity score estimated as before  $\delta(\mathbf{b}_i, \mathbf{d}_j) = 1 - \frac{1}{r} \text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$ .

The training settings are dependent on the recommendation settings, i.e, warm-start, cold-start item, and cold-start user.  $\mathcal{L}_{CGH}(\Theta, \Phi)$  aims to minimize the rating loss and two content reconstruction errors with regularizers. (a.) For the warm-start recommendation, ratings for all users and items are available, then the above objective is trivially optimized by setting the content weights to 0 and learning hashing function with the observed ratings  $\mathbf{R}$ . (b.) For the cold-start item recommendation, ratings for some items are missing, then the objective is optimized by setting the user content weight to 0 and learning parameters with the observed ratings  $\mathbf{R}$  and item content  $\mathbf{V}$ . (c.) The training setting for the cold-start user recommendation is similar to the cold-start item recommendation.

### 3 EXPERIMENTS

We validate the proposed CGH on two public dataset: CiteUlike<sup>1</sup> and RecSys 2017 Challenge dataset<sup>2</sup> from the following two aspects:

- (1) **Marketing analysis.** To validate the effectiveness of CGH in marketing area, we first defined a metric to evaluate the accuracy of mining potential users; we then test the performance for warm-start item and cold-start item, respectively.
- (2) **Recommendation performance.** We test the performance of CGH for recommendation in various settings including: warm-start, cold-start item, and cold-start user in terms of Accuracy@k (Yin et al., 2014).

In the following, we first introduce the experimental settings, followed by the experimental results analysis from the above aspects.

#### 3.1 EXPERIMENTAL SETTINGS

To evaluate the power of finding out potential users and the accuracy of recommendation in different settings. (1) For the CiteUlike dataset, it contains 5,551 users, 16,980 articles, 204,986 observed user-article binary interaction pairs, and articles abstract content. Similar to (Wang & Blei, 2011), we extract bag-of-the-words item vector with dimension  $d_v = 8000$  by ranking the TF-IDF values. (2) For the RecSys 2017 Challenge dataset, it is the only publicly available datasets that contains both user and item content data enabling both cold-start item and cold-start user recommendation. It contains 300M user-item interactions from 1.5M users to 1.3M items and content data collected from the career oriented social network XING (European analog of LinkedIn). Like (Volkovs et al., 2017), we evaluate all methods on binary rating data, item content with dimension of  $d_u = 831$  and user content with the dimension of  $d_v = 2738$ . user features and 2738 item features forming the dimensions of user and item content .

We randomly split the binary interaction (rating)  $\mathbf{R}$  into three disjoint parts: warm start ratings  $\mathbf{R}^w$ , cold-start user ratings  $\mathbf{R}^u$ , and cold-start item ratings  $\mathbf{R}^v$ , and  $\mathbf{R}^w$  is furtherly split into the training dataset  $\mathbf{R}^{wt}$  and the testing dataset  $\mathbf{R}^{we}$ . Correspondingly, the user and item content datasets are split into three disjoint parts. The randomly selection is carried out 5 times independently, and we report the experimental results as the average values.

#### 3.2 EVALUATION METRIC

The ultimate goal of recommendation is to find out the top- $k$  items that users may be interested in. Accuracy@ $k$  was widely adopted by many previous ranking based recommender systems (Koren, 2008; Chen et al., 2009). Thus we adopt the ranking-based evaluation metric Accuracy@ $k$  to evaluate the quality of the recommended item ranking list.

**Metric for Marketing Application.** For a new application of the recommender system, there haven't yet a metric to evaluate the marketing performance. Thus, in this paper, we define an evaluation metric similar to the ranking-based metric Accuracy@ $k$  used for the warm-start and cold-start recommendation in this paper.

From Fig. 2, we discover the  $k$  nearest potential users for an item  $j$ . The basic idea of the metric is to test whether the user that really interested in an item appears in the  $k$  potential users list. For each positive rating ( $r_{ij} = 1$ ) in the testing dataset  $D_{test}$ : (1) we randomly choose 1000 negative users (users  $k$  with  $r_{kj} = 0$ ) and find  $k$  potential users in the 1001 user set; (2) we check if the positive user  $i$  (with positive rating  $r_{ij} = 1$ ) appears in the  $k$  potential users list. If the answer is 'yes' we have a 'hit' and have a 'miss' otherwise.

The metric also denoted by Accuracy@ $k$  is formulated as:

$$\text{Accuracy@k} = \frac{\#hit@k}{|D_{test}|}, \tag{12}$$

<sup>1</sup><http://www.citeulike.org/faq/data.adp>

<sup>2</sup><http://www.recsyschallenge.com/2017/>

where  $|D_{test}|$  is the size of the test set, and  $\#hit@k$  denotes the number of hits in the test set.

### 3.3 ACCURACY FOR MINING POTENTIAL USERS

The experiments evaluate the performance of the marketing application in mining potential users for warm-start items on the test dataset  $R^{we}$  and cold-start items on  $R^v$ . Specifically, we first train the model on the training dataset  $R^{wt}$  and the corresponding user and item content data. When the training is completed, we fix parameters and obtain hash codes  $b_i$  and  $d_j$  by making forward passes. Then we generate  $k$  potential users for items in the test dataset by the procedure demonstrated in Fig. 2, and evaluate the quality of the potential users list by Accuracy@k defined in Section 3.2. The

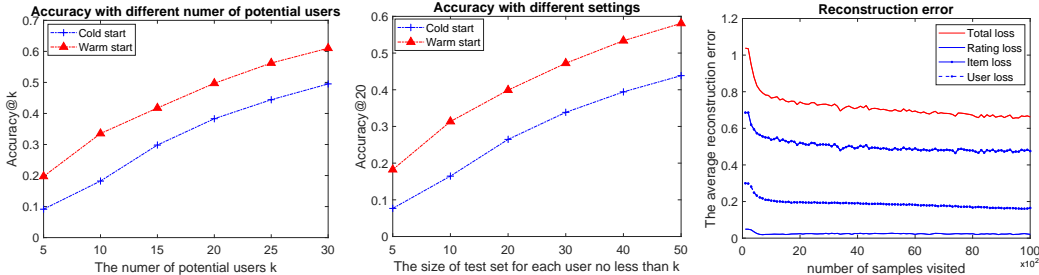


Figure 3: **Left:** The accuracy variation for warm-start and cold-start with the number of potential users. **Center:** The accuracy variation of mining 20 potential users with the number of users (size of the test set) who really interested in the target item. **Right:** The average reconstruction error with the number of visited samples. The total error is the sum of the rating reconstruction error, the user content reconstruction error, and the item reconstruction error.

marketing analysis for warm start item and cold-start item are reported in Fig. 3 (Left.), which shows the accuracy values varies with the numbers of potential users. It indicates the accuracy increases with the number of potential users for both cold-start and warm start settings. It’s reasonable because mining more potential users will have greater accuracy value defined in Section 3.2. Especially, the proposed CGH is effective for cold-start item, which indicates further e-commerce strategies can be developed for new items to attract those potential users. Besides, from the perspective of marketing, warm-start recommendation and cold-start recommendation has less gap than traditional recommendation.

**Robust Testing.** We evaluate the performance varies with the number of users who really interested in the target item in test set. The experimental results shown in Fig. 3 (Center.) indicates the accuracy grows steadily with the size of test set, which reveals the CGH for marketing application is robust. Thus, it is practical to be used in the sparse and cold-start settings.

**Convergence of CGH.** Fig. 3 (Right.) demonstrates the convergence of the proposed CGH, which reveals the reconstruction errors of ratings, users content, items content and the total error with the number of samples seen by CGH are converged, which furtherly validate the correction and effectiveness of the proposed CGH.

### 3.4 ACCURACY FOR RECOMMENDATION

**Accuracy for warm-start Recommendation.** Fig. 4 (Left.) shows the accuracy comparison of warm-start recommendation on CiteUlike dataset. In which collaborative generated embedding (CGE) denotes the real version of CGH. The figure shows the proposed CGH (CGE) has a comparable performance with other hybrid recommender systems. The proposed CGH is hashing-based recommendation, hence binary vectors apply to recommendation which has the advantage in online recommendation as introduced in Section 1; while the baselines are real-valued recommendations which conducts recommendation on real latent space. Due to real latent vectors intuitively carried more information than hash codes. Thus it is acceptable to have small gaps between the real-valued hybrid recommendation and the hashing-based recommendation. In addition, there is still small gap of the real version CGE in comparison with DropoutNet, because the reconstruction error is consid-

ered in CGH(CG), while DropoutNet didn't consider it. However, the reconstruction is significant in the generative step of CGH, which makes it feasible to mining effective potential users, thus CGH(CG) has the advantage in marketing application.

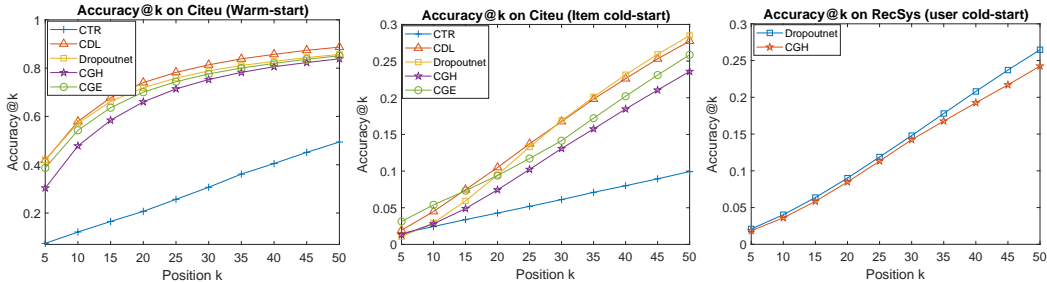


Figure 4: **Left:** The Accuracy variation with the number of recommended items for warm-start recommendation on CiteUlike. **Center:** The Accuracy variation with the number of recommended items for cold-start item recommendation on CiteUlike. **Right:** The Accuracy variation with the number of recommended items for cold-start user recommendation on RecSys

**Accuracy for cold-start item recommendation.** This experiment studies the accuracy comparison between competing hybrid recommender systems and CGH under the same cold-start item setting. We test the performance on the test dataset  $R_v$  introduced in Section 3.1. Specifically, in  $R_v$  each item (cold-start item) has less than 5 positive ratings. Then we select users with at least one positive rating as test users. For each test user, we first choose his/her ratings related to cold-start items as the test set, and the remaining ratings as the training set. Our goal is to test whether the marked-off cold-start items can be accurately recommended to the right user.

The experimental results for cold-start item recommendation are shown in Fig. 4 (**Center**). We conclude that CGH has a comparable performance with competing baselines and achieves better performance than CTR. The results evaluated by another metric MRR (detailed in Appendix.A) are similar.

**Accuracy for cold-start user recommendation.** We also test the performance on the cold-start user setting on the test dataset  $R_u$  introduced in Section 3.1. Specifically, in  $R_u$ , each user (cold-start user) has less than 5 positive ratings. Then we select items with at least one positive rating as test items. For each test item, we first choose ratings related to cold-start users as the test set, and the remaining ratings as the training set. Our goal is to test whether the test item can be accurately recommended to marked-off user.

Due to the fact that only Dropout can be applied to cold-start user recommendation, so we only compare the performance of CGH with Dropout. The experimental results for cold-start user recommendation shown in Fig. 4 (**Right**) indicates our proposed CGH has similar performance with DropoutNet. Besides, CGH has the advantage of the application in marketing area.

## 4 CONCLUSION

In this paper, a generated recommendation framework called collaborative generated hashing (CGH) is proposed to address the cold-start and efficiency issues for recommendation. The two main contributions are put forward in this paper: (1) we develop a collaborative generated hashing framework with the principle of Minimum Description Length together(MDL) with uncorrelated and balanced constraints on the inference process to derive compact and informative hash codes, which is significant for the accuracy of recommendation and marketing; (2) we propose a marketing strategy by the proposed CGH, specifically, we design a framework to discover the  $k$  potential users by the generate step; (3) we evaluate the proposed scheme on two the public datasets, the experimental results show the effectiveness of the proposed CGH for both warm-start and cold-start recommendation.



## REFERENCES

- Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 91–98. ACM, 2015.
- Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th international conference on World wide web*, pp. 681–690. ACM, 2009.
- Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. Stochastic generative hashing. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 913–922. JMLR. org, 2017.
- Ruining He and Julian McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434. ACM, 2008.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
- Defu Lian, Rui Liu, Yong Ge, Kai Zheng, Xing Xie, and Longbing Cao. Discrete content-aware matrix factorization. In *Proceedings of KDD’17*, pp. 325–334. ACM, 2017.
- Chenghao Liu, Xin Wang, Tao Lu, Wenwu Zhu, Jianling Sun, and Steven CH Hoi. Discrete social recommendation. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.
- Dominik Papies, Peter Ebbes, and Harald J Van Heerde. Addressing endogeneity in marketing models. In *Advanced methods for modeling markets*, pp. 581–627. Springer, 2017.
- Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, pp. 139–146. ACM, 2012.
- Kosetsu Tsukuda, Satoru Fukayama, and Masataka Goto. Abcprec: Adaptively bridging consumer and producer roles for user-generated content recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1197–1200. ACM, 2019.
- Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems*, pp. 4957–4966, 2017.
- Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pp. 77–82, 1999.
- Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 448–456. ACM, 2011.
- Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1235–1244. ACM, 2015.
- Haoyu Wang, Nan Shao, and Defu Lian. Adversarial binary collaborative filtering for implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5248–5255, 2019.
- Qifan Wang, Bin Shen, Zhiwei Zhang, and Luo Si. Sparse semantic hashing for efficient large scale similarity search. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1899–1902. ACM, 2014.

- Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.
- Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Zi Huang. A temporal context-aware model for user behavior modeling in social media systems. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 1543–1554. ACM, 2014.
- Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. Discrete collaborative filtering. In *Proceedings of SIGIR’16*, volume 16, 2016.
- Yan Zhang, Haoyu Wang, Defu Lian, Ivor W Tsang, Hongzhi Yin, and Guowu Yang. Discrete ranking-based matrix factorization with self-paced learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2758–2767. ACM, 2018.
- Zhiwei Zhang, Qifan Wang, Lingyun Ruan, and Luo Si. Preference preserving hashing for efficient recommendation. In *Proceedings of SIGIR’14*, pp. 183–192. ACM, 2014.
- Feng Zheng, Cheng Deng, and Heng Huang. Binarized neural networks for resource-efficient hashing with minimizing quantization loss.
- Ke Zhou and Hongyuan Zha. Learning binary codes for collaborative filtering. In *Proceedings of KDD’12*, pp. 498–506. ACM, 2012.

## APPENDIX

### A. MRR RESULTS ON FOR RECOMMENDATION

We evaluate the accuracy in terms of the MRR (Yin et al., 2014) metric shown in Table 1 for warm-start recommendation. Our proposed CGH performs almost as well as the best result of the real-valued competing baselines. Table 1 summarizes MRR results for the four algorithms, the best result is marked as ‘\*q’ and the second best is marked as ‘oq’. We find that the performance of CGH is very close to the best result, that is consistent with the outcome of Accuracy@k reported in Fig. 4.

Table 1: MRR on CiteUlike

Method	CTR	CDL	Dropoutnet	CGH
‘Warm-startq	0.0324	0.0667*	0.0580	0.0595 <sup>o</sup>
‘Cold-startq	0.0101	0.0150	0.0179*	0.0165 <sup>o</sup>