

Department: Head
Editor: Name, xxxx@email

Collaborative Generative Hashing for Marketing and Fast Cold-start Recommendation

Yan Zhang

University of Electronic Science and Technology of China
Center for Artificial Intelligence, University of Technology Sydney

Ivor W. Tsang

Center for Artificial Intelligence, University of Technology Sydney

Lixin Duan

University of Electronic Science and Technology of China

Abstract—Cold-start has being a critical issue in recommender systems with the explosion of data in e-commerce. Most existing studies proposed to alleviate the cold-start problem are also known as hybrid recommender systems that learn representations of users and items by combining user-item interactive and user/item content information. However, previous hybrid methods regularly suffered poor efficiency bottlenecking in online recommendations with large-scale items, because they were designed to project users and items into continuous latent space where the online recommendation is expensive. To this end, we propose a collaborative generated hashing (CGH) framework to improve the efficiency by denoting users and items as binary codes, then fast hashing search techniques can be used to speed up the online recommendation. In addition, the proposed CGH can generate potential users or items for marketing application where the generative network is designed with the principle of Minimum Description Length (MDL), which is used to learn compact and informative binary codes. Extensive experiments on two public datasets show the advantages for recommendations in various settings over competing baselines and analyze its feasibility in marketing application.

■ **NEW USER OR NEW ITEM RECOMMENDATION** With the development of e-

commerce, we are accustomed to receiving a variety of things we might be seeking, such as movies, books, news, hotels, etc. In the digi-

tal ‘world’ where they are also referred to as items to be (not) recommended. Recommender systems help users find their desirable items, creating new revenue opportunities for vendors, such as Amazon, Taobao, eBay, etc. Traditional recommender systems promote items based on their content similarities, which is also known as content filtering. The drawback of content filtering is its lack of personalized information, which requires to extract features from content information that can represent personalized information to conduct recommendation. Thus, the performance of content filtering depends on the feature learning technique. Another state-of-the-art recommendation framework is collaborative filtering where matrix factorization (MF) has been applied successfully in various scenarios. Collaborative filtering produces recommendations from user-item past interactive information. Most current recommender systems are designed according to MF. However, new items and users make the matrix factorization invalid due to the lack of interactive data, which brings a big challenge to recommender systems. Bottlenecks caused by new items or users are known as cold-start issues in recommender systems.

To solve cold-start issues, many scholars put forward hybrid recommender systems by incorporating ratings with side information, such as image, context, profile, social networks, etc. These hybrid recommendation frameworks first learn user/item representations; and then pass them into the rating prediction model (e.g. the inner product of user/item representations) to predict rating scores; and finally recommend top- k items by ranking these scores. However, most existing hybrid recommender systems formulated in continuous latent space, which leads to low efficiency in the stage of online recommendation when the data size increases. Specifically, suppose the size of the item set is m , the time complexity of recommending top- k items for a specific user in continuous space is $\mathcal{O}(mk+m\log k)$ [1], which is a critical efficiency bottleneck with the growth of items, such as those in Taobao. It’s challenging to meet the fast response requirement for a specific user.

Hashing has being a promising approach to solving the efficiency issue. To learn hash codes of users and items, existing hashing based recom-

mendations have to optimize a nonlinear discrete problem which has proven to be NP-hard [2]. So they transformed the original discrete optimization problem into tractable approximate surrogate problems. There are two types of approximate methods: two-stage hashing and learning-based hashing. Two-stage hashing consists of a continuous latent representation learning by discarding discrete constraints and a simple quantization method like a threshold function, which causes great information loss. Learning-based hashing obtains binary codes by optimizing approximate surrogate mixed-integer problems directly [1], which often achieves better solution for the original discrete problem, but the training of the mixed-integer problems is also expensive.

To solve the above issues, we create hash functions by a collaborative generative network which can speed up the training procedure and reduce the quantization loss. Thus, in this paper we propose a collaborative generated hashing (CGH) to learn hash functions of users and items from content data with the principle of Minimum Description Length (MDL) that has been successfully applied to generate effective hash codes [3].

For new users and new items, manufacturers often do some marketing analysis before expanding or producing a product. A solid marketing strategy is essential to the success of manufacturers. From the perspective of recommendation, the marketing process can be regarded as a cold-start item recommendation. Mining potential users corresponds to recommending the new item to some specific users. Thus, in our work we propose an approach to improve marketing by discovering potential users via 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. Specifically, for a new item, we generate a new potential user with the generative step, and then we search the nearest potential users in the user set. By recommending a new product to potential users who might be interested in but didn’t plan to buy, which is expected to attract those potential users, and thus it’s a promising marketing strategy.

We organize the paper as follows: we first introduce some work closely related to CGH, and follow by introducing the framework of CGH and

comparing it with related competing baselines: CDL [4] and DropoutNet [5]; we then formulate the generative step and illustrate its application in mining potential customers, leading on to the introduction of the inference step, i.e., building hash functions; we finally summarize the training objective and the optimization method, followed by the experimental analysis and the conclusion.

The main contributions of this paper are summarized as follows:

- (1) We propose a Collaborative Generated Hashing (CGH) with the principle of MDL to learn compact but informative hash codes. This applies to various settings for recommendation.
- (2) We provide a marketing strategy by discovering potential users by the generative step of CGH.
- (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 public datasets.

RELATED WORK

Existing studies for solving cold-start problems were mostly modeled as a combination of collaborative filtering and content filtering, known as hybrid recommender systems [6], [4], [7], [5], [8], [9]. Specifically, they learned continuous latent factors by incorporating side information into the interactive data – such as Collaborative Deep Learning (CDL) [4], Visual Bayesian Personalized Ranking (VBPR) [7], Collaborative Topic modeling for Recommendation (CTR) [6] and the DropoutNet for addressing cold start issues [5].

Hashing techniques applied for speeding up the online recommendation fall into two groups: two-stage hashing and learning-based hashing. Two classic two-stage hashing methods are preference preserving hashing (PPH) [10] and iterative quantization (ITQ) [11]. Due to substantial information loss caused by the quantization stage, learning-based hashing frameworks were proposed to learn hash codes directly, and are composed of two types: bit-wise learning, such as discrete collaborative filtering (DCF) [12] and block-wise learning [13], [14]. In addition, studies of hashing method also appears in other fields, such as cycle-consistent deep generative hashing for cross-modal retrieval [15], generative

reconstructive hashing for Video Analysis [16], deep semantic hashing with generative adversarial networks [17], etc. These generative hashing methods also motivate us to propose a generative hashing model for recommendation.

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 (hash 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 generate a new user via the generative step for an item to search potential users who would be interested in the item.

To distinguish our proposed framework from existing associated methods, we have drawn a diagramme shown in Fig. 1 to clarify our contributions by comparing of the collaborative deep learning (CDL) [4] and the DropoutNet [5]. In contrast to CDL, the proposed CGH aims to learn binary codes instead of continuous latent vectors P and Q because of the advantage of hashing for online recommendation; thus, the CGH optimizes an objective with the principle of MDL, while CDL optimizes the joint objective of rating loss and item content reconstruction error. In comparison with DropoutNet, CGH can be applied into marketing by discovering potential users; besides, CGH learns hash codes by the stacked denoising auto-encoder, while DropoutNet obtains continuous latent factors from the standard neural network.

Stacked denoising auto-encoders can reconstruct data from an input of corrupted data. After giving corrupted data to the auto-encoder. It forces the hidden layer to learn robust representations; then the output will be a refined version of the input data [18]. Stacked denoising auto-encoders solve this problem by corrupting the data on purpose by randomly turning some of the input values to zero. In general, the percentage of input nodes which are being set to zero depends

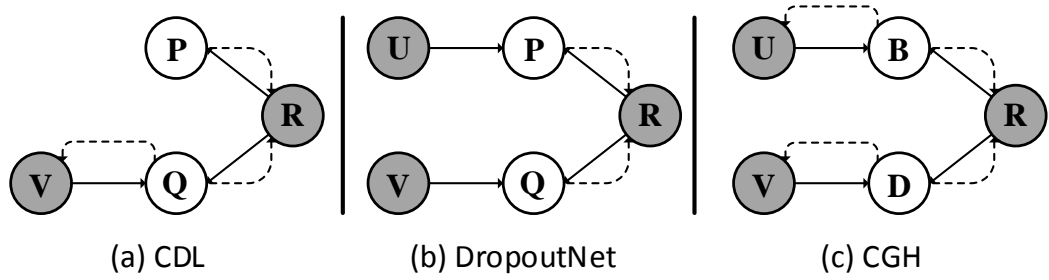


Figure 1: Differences between CDL, DropoutNet and our proposed CGH. Solid lines and dashed lines respectively represent the inference (encoding) and the generative (decoding) process. The shaded nodes U, V, R are observed user content, item content and rating, respectively. $P(B), Q(D)$ denotes continuous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

on the amount of data and input nodes we have.

We first formulate the inference step – constructing hash functions; we then formulate the generative process and demonstrate the marketing application – mining potential customers; finally, we summarize the training objective and the optimization method.

Hash Functions

The inference process (also known as encoding process) shown in the Fig. 1 (c) with solid lines, the binary latent variables b_i (d_j) depends on the content vector u_i (v_j) and the rating R . Inspired by the recent work on generative hashing [3] and DropoutNet [5], we use a multivariate Bernoulli distribution to model the inference process of b_i and d_j with linear parametrization, i.e.,

$$\begin{aligned} q(b_i|\tilde{u}_i) &= \mathcal{B}(\sigma(\mathcal{T}_u^T \tilde{u}_i)) \\ q(d_j|\tilde{v}_j) &= \mathcal{B}(\sigma(\mathcal{T}_v^T \tilde{v}_j)), \end{aligned} \quad (1)$$

where $\tilde{u}_i = [u_i, p_i]$, $\tilde{v}_j = [v_j, q_j]$, $\sigma(x) = (1 + e^{-x})^{-1}$. p_i and q_j are the results of r -dimension matrix factorization [19] of R , i.e., $r_{ij} \approx p_i^T q_j$. $\mathcal{T}_u = [t_{uk}]_{k=1}^r$, $t_{uk} \in \mathbb{R}^{d_u+r}$, $\mathcal{T}_v = [t_{vk}]_{k=1}^r$, $t_{vk} \in \mathbb{R}^{d_v+r}$ are the output of the hidden layer in the stacked auto-encoder applied in this paper. From SGH [3], the MAP solution of the eq. (1) is readily given by

$$\begin{aligned} b_i &= \underset{b_i}{\operatorname{argmax}} q(b_i|u_i) = \frac{\operatorname{sign}(\mathcal{T}_u^T u_i) + 1}{2}, \\ d_j &= \underset{d_j}{\operatorname{argmax}} q(d_j|v_j) = \frac{\operatorname{sign}(\mathcal{T}_v^T v_j) + 1}{2}. \end{aligned} \quad (2)$$

With the linear projection followed by a sign function, we can easily get hash codes of users

and items. On this basis, we also call the inference step the construction of hash functions.

Mining Potential Customers

Given a sparse rating matrix R and item content data $V \in \mathbb{R}^{d_v}$, where d_v is the dimension of content vectors (bag-of-words vectors), and V is stacked by the bag-of-words vectors of item content in the item set V . Most previous studies focus on modeling deterministic frameworks to learn representations of items, such as CDL, CTR, DropoutNet, etc. 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. Inference applies knowledge from a trained neural network model, and we use it to infer a result. In this paper, after training we obtain the trained stacked denoising auto-encoder, and then we infer items' binary representations from item content information, and we called the process of inferring as inference step or encoding process. After the inference step, the binary code of item j is available. By maximizing the similarity function $\delta(b_i, d_j)$ (also known as preference predicted model in collaborative filtering), the optimal binary code b_p is obtained.

Given a latent representation, we can reconstruct the corresponding input by the decoding process(also called 'generative step'). Thus, we generate a new user u_p from the binary code b_p via the generative step. Finally we discover potential users from the user set with the aid of this new user by some nearest neighborhood

algorithms such as KNN. As a marketing strategy, the generative step can detect potential users for both warm-start and cold-start items. From the perspective of marketing, it can be regarded as another kind of item recommendation.

The generative process is denoted by dashed lines in Fig. 1 (c). Fix binary codes \mathbf{b}_i and \mathbf{d}_j , the bag-of-words vectors \mathbf{u}_i and \mathbf{v}_j are generated via the generative network $p(\theta_u)$ and $p(\theta_v)$, respectively. The ratings r_{ij} is generated by the similarity function of \mathbf{b}_i and \mathbf{d}_j . We use normal Gaussian distribution to model the generation of \mathbf{u}_i and \mathbf{v}_j given \mathbf{b}_i and \mathbf{d}_j , respectively:

$$\begin{aligned} p(\mathbf{u}_i|\mathbf{b}_i) &= \mathcal{N}(\mathbf{C}_u \mathbf{b}_i, \lambda_u^{-1} \mathbf{I}), \\ p(\mathbf{v}_j|\mathbf{d}_j) &= \mathcal{N}(\mathbf{C}_v \mathbf{d}_j, \lambda_v^{-1} \mathbf{I}), \end{aligned} \quad (3)$$

where $\mathbf{C}_{uk} = [\mathbf{c}_{uk}]_{k=1}^r$, $\mathbf{c}_{uk} \in \mathbb{R}^{d_u}$ is the codebook [3] with r codewords. The definition of \mathbf{C}_v is similar to \mathbf{C}_u . d_u is the dimension of the user content vector. The prior is modeled as the multivariate Bernoulli distribution on hash codes: $p(\mathbf{b}_i) \sim \mathcal{B}(\rho_u)$, and $p(\mathbf{d}_j) \sim \mathcal{B}(\rho_v)$. So, the prior probability is as follows:

$$\begin{aligned} p(\mathbf{b}_i) &= \prod_{k=1}^r \rho_{uk}^{b_{ik}} (1 - \rho_{uk})^{1-b_{ik}}, \\ p(\mathbf{d}_j) &= \prod_{k=1}^r \rho_{vk}^{d_{jk}} (1 - \rho_{vk})^{1-d_{jk}}, \end{aligned} \quad (4)$$

We predict the rating by the similarity of binary codes \mathbf{b}_i and \mathbf{d}_j with $\delta(\mathbf{b}_i, \mathbf{d}_j) = 1 - \frac{1}{r} \text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$, where $\text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$ represents the Hamming distance between hash codes, which has been extensively applied for the preference prediction in hashing-based recommender system [6], [20], [13]. Then, the rating is thus drawn from the normal distribution centered at the predicted rating score,

$$p(r_{ij}|\mathbf{b}_i, \mathbf{d}_j) \sim \mathcal{N}(\delta(\mathbf{b}_i, \mathbf{d}_j), C_{ij}^{-1}), \quad (5)$$

where C_{ij} is the precision parameter that serves as confidence of r_{ij} , which is similar to that in CTR [6] ($C_{ij} = a$ 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

$$\begin{aligned} p(\mathbf{R}, \mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{D}) &= \prod_{i,j} p(r_{ij}|\mathbf{b}_i, \mathbf{d}_j) \\ &\quad p(\mathbf{u}_i|\mathbf{b}_i) p(\mathbf{v}_j|\mathbf{d}_j) p(\mathbf{b}_i) p(\mathbf{d}_j) \end{aligned} \quad (6)$$

Model Training

Since our goal is to reconstruct users, items and ratings by using the least information of binary codes. Thus, we train the CGH with the MDL principle, which finds the best parameters for generating hash codes that maximally compress the training data while keeping the information carried. So, CGH aims to minimize the expected amount of information related to q :

$$\mathcal{L}(q) = E_q[\log p(\mathbf{R}, \mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{D}) - \log q(\mathbf{B}, \mathbf{D})] \quad (7)$$

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

$$\begin{aligned} \mathcal{L}_{MAP}(\Theta, \Phi) &= - \sum_{i,j} \frac{C_{ij}}{2} (r_{ij} - \delta(\mathbf{b}_i, \mathbf{d}_j))^2 \\ &\quad - \frac{\lambda_u}{2} \sum_i (\mathbf{u}_i - \mathbf{C}_u \mathbf{b}_i)^2 - \frac{\lambda_v}{2} \sum_j (\mathbf{v}_j - \mathbf{C}_v \mathbf{d}_j)^2 \\ &\quad - \text{KL}(q_{\phi_u} || p_{\theta_u}) - \text{KL}(q_{\phi_v} || p_{\theta_v}) - \nabla(\Theta, \Phi) \end{aligned} \quad (8)$$

where $\Theta = \{\theta_u, \theta_v\}$, $\Phi = \{\phi_u, \phi_v\}$, $\nabla(\Theta, \Phi)$ is the regularizer term of parameters Θ and Φ .

We train all components jointly by back-propagation. After training, we fix all parameters and make forward passes to map content data $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ to binary codes \mathbf{B} and \mathbf{D} , respectively. The recommendation in various settings is then performed using \mathbf{B} and \mathbf{D} in Hamming space by ranking the preference predicted scores (similarity function) $\delta(\mathbf{b}_i, \mathbf{d}_j) = 1 - \frac{1}{r} \text{Hamdis}(\mathbf{b}_i, \mathbf{d}_j)$.

The training settings depend on the recommendation settings, i.e., the warm-start setting, the cold-start item setting, and the cold-start user setting. Specifically, $-\mathcal{L}_{MAP}(\Theta, \Phi)$ aims to minimize the rating loss and two content reconstruction errors with regularizers. (a.) For the warm-start recommendation, less than 1 rating for all users and items are not available. Then, the above objective is optimized trivially by setting the content weights to 0 and learning hashing function with the observed ratings \mathbf{R} . (b.) For

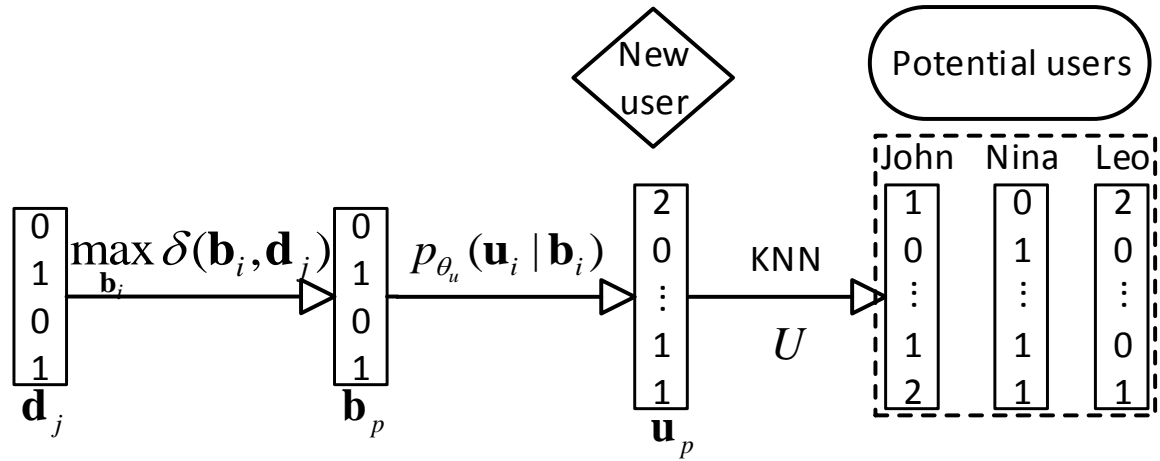


Figure 2: Demonstration of mining potential users for an item j . After the inference step, \mathbf{d}_j is available, we first find out the most similar binary code \mathbf{b}_p ; we then generate a new potential user \mathbf{u}_p by the generative process; we further search the top- k nearest potential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

the cold-start item recommendation, ratings for cold-start items are missing in the training set. In this case, 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 that for the cold-start item recommendation.

RESULTS

We evaluate the proposed CGH on two public datasets: CiteUlike¹ and RecSys 2017 Challenge dataset² from the following three aspects:

- (1) *Online Recommendation Efficiency.* Once training is completed, we test the online recommendation efficiency respectively in terms of real-based hybrid frameworks and hashing-based frameworks.
- (2) *Marketing Analysis.* To validate the feasibility of CGH in marketing area, we first define a metric to evaluate the accuracy of mining potential users; we then test the performance separately in the warm-start and cold-start settings.
- (3) *Recommendation Accuracy.* We test the accuracy of CGH performance in various settings: the warm-start, the item cold-start, and the user cold-start with the metric Accuracy@k [21].

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

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

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

Experimental Settings

To evaluate the effectiveness of mining 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 [6], we extract bag-of-the-words item vectors with dimension $d_v = 8000$ by ranking the TF-IDF values. (2) The RecSys 2017 Challenge dataset is the only publicly available dataset that contains both user and item content data, enabling both the 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 [5], we evaluate all methods on binary rating data (implicit feedbacks) and item content with the dimension $d_u = 831$, and user content with the dimension $d_v = 2738$.

We randomly split the implicit feedbacks (explicit ratings can be transformed into the interval of $[0, 1]$) \mathbf{R} into three disjoint parts: warm-start ratings \mathbf{R}^w , cold-start user ratings \mathbf{R}^u , and the cold-start item ratings \mathbf{R}^v . The warm-start ratings \mathbf{R}^w is further 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.

In the experiments, we apply the 5-fold cross-validation method on random splits of training data and report the experimental results as the average values. The hyper-parameters settings in CGH is as follows: we set the dimension $r = 50$, the layer structure of inference step is the same with DropoutNet. We set hyper-parameters for baselines according to their papers.

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 [22], [23]. So, we adopt the ranking-based evaluation metric Accuracy@ k to evaluate the quality of the recommended item ranking list.

As a new application of the recommender system, there is not yet a metric to evaluate the marketing performance. Hence, 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 a user in the potential users list is really interested in the item. 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 a ‘miss’ otherwise.

The metric denoted by Accuracy@ k is formulated as:

$$\text{Accuracy@}k = \frac{\#hit@k}{|D_{test}|}, \quad (9)$$

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

Online Recommendation Efficiency

After training, continuous latent vectors and hash codes are respectively obtained by real-based hybrid recommender systems and hashing-based recommendations. We recommend top-10 items for a specific user separately in continuous

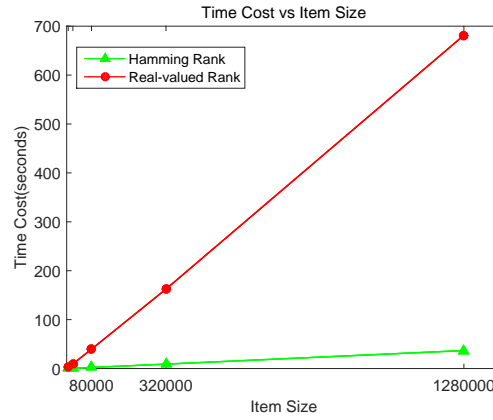


Figure 3: The time efficiency comparison of real-based recommendation frameworks and hashing-based methods.

latent space and Hamming space. We investigate the time cost on synthetic datasets for both two types of methods. We first use standard Gaussian distribution to generate items’ continuous latent vectors randomly. Items hash codes are obtained from these real-valued vectors by the sign function. We set different sizes of items sets in the experiment: 80,000, 320,000, ..., 1,280,000, to test the time cost of online recommendation. We denote the two kinds of online recommendations as ‘real-valued rank’ and ‘Hamming rank’ in Fig. 3. The experimental results tell us recommending in Hamming space is much more efficient than that in real space. In fact, hashing technique has been applied successfully for fast image search. It also works here because both image search and recommendation task are essentially a similarity search.

Marketing Analysis

The experiments consecutively evaluate performances of marketing application by mining potential users under the warm-start and cold-start settings. Specifically, we first train the model with 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 each item in the test dataset (the flowchart is illustrated in Fig. 2) and evaluate the quality of the generated potential users by the metric, ‘Accuracy@ k ’.

The marketing analysis for warm-start items

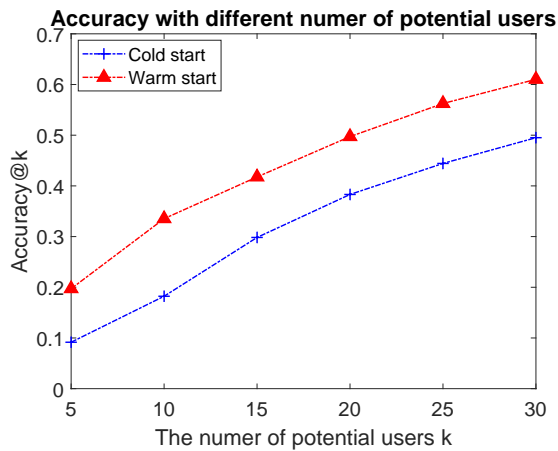


Figure 4: The marketing accuracy testing for the warm-start and cold-start settings

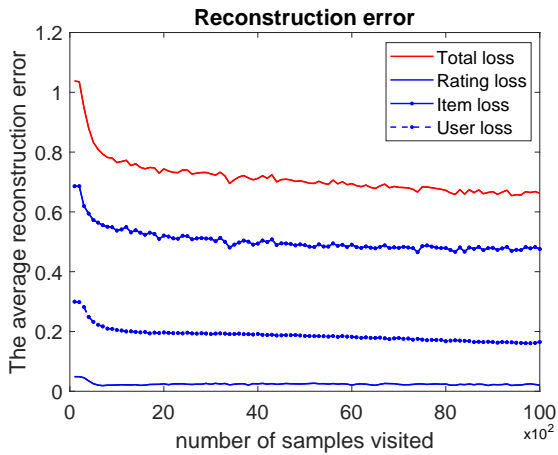


Figure 5: The average reconstruction error with the iteration training step on CiteUlike. The total error is the sum of reconstruction errors for ratings, users' and items' content

and cold-start items are reported in Fig. 4, which displays how the marketing accuracy varies with the number of potential users. It indicates that accuracy increases with the number of potential users for both the cold-start and warm-start settings. It's reasonable because mining more potential users increases the probability to cover the user who are really interested in the item. The accuracy in the cold-start setting is comparable to that in the warm-start setting, so CGH can be used as an approach for marketing. Moreover, for a new item, we can apply CGH to find some potential users.

Fig. 5 demonstrates the convergence of CGH that reveals the training process is converged and correct.

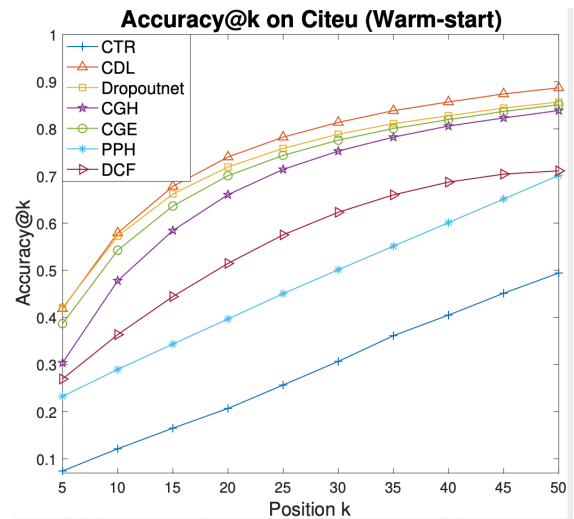


Figure 6: The accuracy comparison with the number of recommended items in the warm-start setting on CiteUlike dataset.

Recommendation Accuracy

We assess the effectiveness of the proposed CGH by testing the accuracy of recommending top- k items in Hamming space. Specifically, after training, we apply hash codes of users and items to conduct online recommendation by fast hashing technique in Hamming space. Please note that this procedure is different from the marketing application, because marketing process can also provide recommendation.

Accuracy for the Warm-start Recommendation. Fig. 6 shows the accuracy comparison with the warm-start setting on CiteUlike dataset. In this figure, collaborative generated embedding (CGE) denotes the continuous version of the proposed CGH. The figure shows the proposed CGH (CGE) has a comparable performance with other hybrid recommender systems, and has the advantage of performance over other two hashing models PPH and DCF. It's worth noting that the proposed CGH is a hashing-based recommendation, and it has the efficiency advantage in online recommendation verified before. Due to continuous latent vectors intuitively carried more information than hash codes. So, it is acceptable to have small gaps between the real-valued hybrid recommendation and the hashing-based recommendation.

In addition, there is still a small gap between CGE and DropoutNet, because the content re-

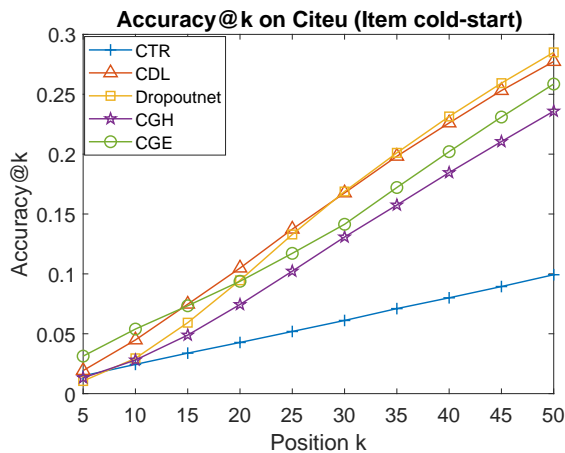


Figure 7: The accuracy variation with the number of recommended items for the warm-start setting on CiteUlike.

construction error is taken into consideration in CGH(CGE); while DropoutNet didn't consider it. In our work, the reconstruction error is significant in the generative step of CGH, which makes it a feasible approach to mining effective potential users for marketing application.

Accuracy for the Cold-start Item Recommendation. We test the recommendation accuracy on the test dataset R_v for the competing hybrid recommender systems and the proposed CGH under the same cold-start setting. Specifically, we first choose items with less than 5 ratings as cold-start items R_v , and the remaining items are regarded as warm-start items; we then train our model with warm-start ratings (i.e., ratings from warm-start users to warm-start items) and the corresponding warm-start items' content data; we finally predict ratings by the obtained users' presentations and parameters of hash functions, and test the performance on the cold-start items R_v . Each item in R_v owes no more than 5 positive ratings, thus these items are treated as cold-start items. 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 the cold-start item recommendation are shown in Fig. 7. We conclude that CGH has a comparable perfor-

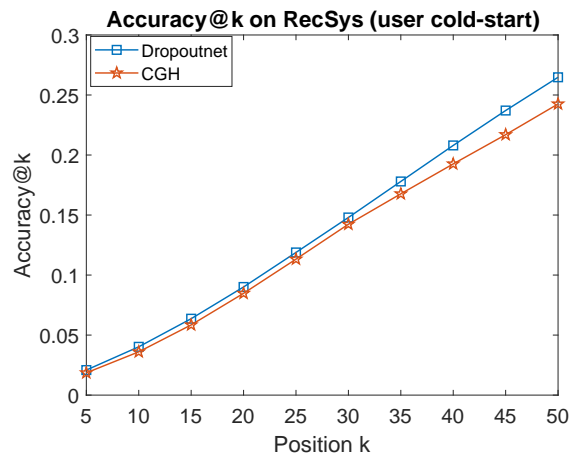


Figure 8: The accuracy variation with the number of recommended items for the warm-start setting on CiteUlike.

mance 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 of the cold-start user setting on the test dataset R_u . Specifically, in R_u , each user (the 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.

Among baselines, only DropoutNet was designed to deal with both the cold-start user and cold-start item recommendations, so, we test the accuracy of DropoutNet and the proposed CGH under the cold-start user setting. The experimental results shown in Fig. 8 indicate our proposed CGH can achieve comparable performance with DropoutNet. Additionally, CGH has efficient advantages in online recommendation and can be applied in marketing area.

Conclusion

In this paper, a generative recommendation framework called collaborative generated hashing (CGH) is proposed to address the cold-start and efficiency issues for recommendation. The two main contributions put forward here are: (1) we

develop a collaborative generated hashing framework with the principle of Minimum Description Length together (MDL) to derive compact and informative hash codes which are essential for the accurate recommendation and marketing; (2) the proposed CGH has meaningful application in marketing area by the generative step. To be precise, we design a framework to discover the k potential customer using the generate step; (3) we evaluate the proposed scheme on two public datasets: the experimental results show we can achieve efficient and accurate online recommendation with hashing under both the warm-start and cold-start settings.

REFERENCES

1. Y. Zhang, D. Lian, and G. Yang, "Discrete personalized ranking for fast collaborative filtering from implicit feedback," in *AAAI*, 2017, pp. 1669–1675.
2. J. Håstad, "Some optimal inapproximability results," *Journal of the ACM (JACM)*, vol. 48, no. 4, pp. 798–859, 2001.
3. B. Dai, R. Guo, S. Kumar, N. He, and L. Song, "Stochastic generative hashing," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 913–922.
4. H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1235–1244.
5. M. Volkovs, G. Yu, and T. Poutanen, "Dropoutnet: Addressing cold start in recommender systems," in *Advances in Neural Information Processing Systems*, 2017, pp. 4957–4966.
6. C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 448–456.
7. R. He and J. McAuley, "Vbpr: visual bayesian personalized ranking from implicit feedback," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
8. S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. Association for the Advancement of Artificial Intelligence, 2018, pp. 2532–2539.
9. S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 6332–6338.
10. Z. Zhang, Q. Wang, L. Ruan, and L. Si, "Preference preserving hashing for efficient recommendation," in *Proceedings of SIGIR'14*. ACM, 2014, pp. 183–192.
11. K. Zhou and H. Zha, "Learning binary codes for collaborative filtering," in *Proceedings of KDD'12*. ACM, 2012, pp. 498–506.
12. H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *Proceedings of SIGIR'16*, vol. 16, 2016.
13. Y. Zhang, H. Wang, D. Lian, I. W. Tsang, H. Yin, and G. Yang, "Discrete ranking-based matrix factorization with self-paced learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2758–2767.
14. H. Wang, N. Shao, and D. Lian, "Adversarial binary collaborative filtering for implicit feedback," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5248–5255.
15. L. Wu, Y. Wang, and L. Shao, "Cycle-consistent deep generative hashing for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1602–1612, 2018.
16. J. Zhang, Z. Wei, I. C. Duta, F. Shen, L. Liu, F. Zhu, X. Xu, L. Shao, and H. T. Shen, "Generative reconstructive hashing for incomplete video analysis," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 845–854.
17. Z. Qiu, Y. Pan, T. Yao, and T. Mei, "Deep semantic hashing with generative adversarial networks," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 225–234.
18. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.
19. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.
20. D. Lian, R. Liu, Y. Ge, K. Zheng, X. Xie, and L. Cao, "Discrete content-aware matrix factorization," in *Proceedings of KDD'17*. ACM, 2017, pp. 325–334.
21. H. Yin, B. Cui, L. Chen, Z. Hu, and Z. 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*. ACM, 2014, pp. 1543–1554.

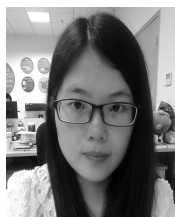
22. Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.
23. W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang, “Collaborative filtering for orkut communities: discovery of user latent behavior,” in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 681–690.

APPENDIX: ANOTHER METRIC EXPERIMENTS

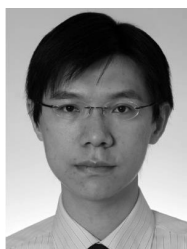
We evaluate the accuracy in terms of the MRR [21] metric shown in Table 1 for the warm-start recommendation. Our proposed CGH performs as well as the best result of the real-valued competing baselines. Table 1 summarizes MRR results for the four algorithms: the best result is marked as ‘★’ and the second best is marked as ‘*o*’. We find that the performance of CGH is very close to the best result, which is consistent with the outcome of Accuracy@*k* reported in Fig. 6.

Table 1: MRR on CiteUlike

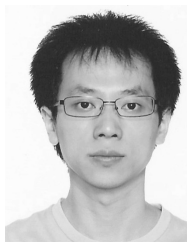
Method	CTR	CDL	Dropoutnet	CGH
‘Warm-start’	0.0324	0.0667★	0.0580	0.0595 ^o
‘Cold-start’	0.0101	0.0150	0.0179★	0.0165 ^o



[] Yan Zhang is currently working as a Postdoc in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Her research interests include recommender systems, data mining, machine learning, etc.



[] Ivor W. Tsang is an ARC Future Fellow and Professor of Artificial Intelligence, at University of Technology Sydney (UTS). He is also the Research Director of the Australian Artificial Intelligence Institute (AAIL). He has 200 research papers published in top-tier journals and conferences. According to Google Scholar, he has more than 15,000 citations and his H-index is 57. In 2013, Prof Tsang received his prestigious Australian Research Council Future Fellowship for his research regarding Machine Learning on Big Data. In 2019, his Journal of Machine Learning Research paper titled “Towards ultrahigh dimensional feature selection for big data” received the International Consortium of Chinese Mathematicians Best Paper Award. In 2020, Prof Tsang was recognized as the AI 2000 AAAI/IJCAI Most Influential Scholar in Australia for his outstanding contributions to the field of AAAI/IJCAI between 2009 and 2019. In addition, he had received the IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2007, the 2014 IEEE Transactions on Multimedia Prize Paper Award, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010. He serves as a Senior Area Chair/Area Chair for NeurIPS, ICML, AISTATS and IJCAI. He serves as the Editorial Board for the Journal of Machine Learning Research and Machine Learning Journal, and serves as an Associate Editor for the IEEE Transactions on Big Data and the IEEE Transactions on Emerging Topics in Computational Intelligence.



[] Lixin Duan received the B.Eng. degree from the University of Science and Technology of China in 2008, and the Ph.D. degree from Nanyang Technological University in 2012. He is currently a Full Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). His re-

Department Head

search interests include machine learning algorithms (especially in transfer learning and domain adaptation) and their applications in computer vision. He was a recipient of the Microsoft Research Asia Fellowship in 2009 and the Best Student Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition 2010.