Knowl Inf Syst (2017) 52:221–254
DOI 10.1007/s10115-016-1005-1

REGULAR PAPER

Exploiting interactions of review text, hidden user communities and item groups, and time for collaborative filtering

Yinqing Xu[1] · Qian Yu[1] · Wai Lam[1] · Tianyi Lin[1]

Received: 31 May 2016 / Revised: 5 October 2016 / Accepted: 10 November 2016 /
Published online: 29 November 2016
© Springer-Verlag London 2016

Abstract Rich side information concerning users and items are valuable for collaborative filtering (CF) algorithms for recommendation. For example, rating score is often associated with a piece of review text, which is capable of providing valuable information to reveal the reasons why a user gives a certain rating. Moreover, the underlying community and group relationship buried in users and items are potentially useful for CF. In this paper, we develop a new model to tackle the CF problem which predicts user's ratings on previously unrated items by effectively exploiting interactions among review texts as well as the hidden user community and item group information. We call this model CMR (co-clustering collaborative filtering model with review text). Specifically, we employ the co-clustering technique to model the user community and item group, and each community–group pair corresponds to a co-cluster, which is characterized by a rating distribution in exponential family and a topic distribution. We have conducted extensive experiments on 22 real-world datasets, and our proposed model CMR outperforms the state-of-the-art latent factor models. Furthermore, both the user's preference and item profile are drifting over time. Dynamic modeling the temporal changes in user's preference and item profiles are desirable for improving a recommendation system. We extend CMR and propose an enhanced model called TCMR to consider time information and exploit the temporal interactions among review texts and co-clusters of user communities and item groups. In this TCMR model, each community–group co-cluster is characterized by an additional beta distribution for time modeling. To evaluate our TCMR model, we have conducted another set of experiments on 22 larger datasets with wider time span. Our proposed model TCMR performs better than CMR and the standard time-aware recommendation model on the rating score prediction tasks. We also investigate the temporal effect on the user–item co-clusters.

Keywords Collaborative filtering · Latent factor modeling · Co-clustering

✉ Qian Yu
yuqian@se.cuhk.edu.hk

[1] Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, Hong Kong, SAR, China

المنارة للاستشارات

1 Introduction

Over the years, with the rapid development of e-commerce, the amount of information available on e-commerce websites has become increasingly enormous. Many popular e-commerce websites are looking for advanced technology that can help provide high-quality personalized service to match users with items of interest. To address this problem, recommendation system is widely used for the purpose of producing a list of personalized recommended product items for users based on their historical purchased behaviors [11]. Among various recommendation techniques, collaborative filtering (CF) is the most prominent approach which has been applied on major commercial companies and proven effective in many domains such as movies, books, and food. Typically, CF takes the rating scores of different users for different items as input. Such rating information is commonly represented by a user–item rating matrix. A common scenario is that the elements in the user–item rating matrix are partially filled. The goal is to predict user's ratings on previously unrated items. One technique for tackling this rating prediction problem is to examine the rating behaviors of users who share similar interests or tastes. Koren [33] proposed the latent factor model for solving CF and witnessed some success. Technically each user and each item is modeled as a latent factor vector with the same dimension, and such latent vector can be inferred from the observed rating matrix by means of matrix factorization. The user's rating on an item can be predicted as the inner product between the corresponding user latent factor vector and item latent factor vector.

Most CF algorithms only consider the user–item rating scores as described above, but the rich side information concerning users and items are also valuable for recommendation [50]. In fact, it is often the case that many modern e-commerce sites, such as Amazon, eBay, and Epinions, contain review texts in addition to rating scores. Such review texts can provide us valuable information to explain why the user gives this rating. A rating score can indicate the user's overall satisfaction for the item but cannot reveal the underlying possible rationale. Generally users give the same level of rating due to different reasons. For example, in the domain of hotel, a user may be attracted by the great location and gives a 5-star rating, while another user may give the same 5-star rating just due to the friendly service. Nevertheless, most existing CF approaches [8,21,40,47] for recommendation have ignored this valuable text information.

Moreover, the underlying community and group relationship embedded in the users and items is potentially useful for CF. For example, the recommended items can often be categorized into different hidden groups (e.g., action, adventure, horror movie, and comedy movie), and users can also be identified as different hidden communities. Such underlying relationship can be effectively modeled by co-clustering, which is capable of conducting simultaneous clustering of two variables, to better predict the observed ratings and review texts. In terms of the rating scores, since the input user–item rating matrix is dyadic involving the mutual interactions between users and items, traditional algorithms would not perform well on uncovering the community and group information of each user and each item [49]. On the contrary, co-clustering [28] is able to take advantage of the user–item relationships leading to better prediction of rating scores [2]. Additionally, co-clustering can be more effective for modeling the generation of review texts since different user communities would discuss different topics and vary their own wordings or expression patterns when dealing with different item groups. For example, a user community, who cares more about the style of clothings, tends to use wordings such as "fashionable" and "beautiful" for the T-shirt item group. "durable" and "high quality" would frequently appear in the reviews written by the user community concerning the quality of T-shirts. Recently, McAuley and Leskovec [40]

المنارة للاستشارات

proposed a refined latent factor model named HFT which attain better predictive accuracy for user's ratings on items of potential interest by incorporating the review texts. Almahairi et al. [4] also proposed a neural network-based approach with the consideration of the review texts. However, each user and each item in both of these models is just represented by a latent factor vector, and these latent factors are estimated by approximating the observed rating score with the inner product between the corresponding user latent factor vector and item latent factor vector. As a result, both models do not explicitly consider any underlying user community and item group information. Another limitation of existing latent factor models is that both users and items are characterized by the same number of factors. However, since the complexity embedded in users should vary from that of items, it would lead to overfitting or underfitting problems if we pose an inappropriate constraint for the number of latent factors.

In this paper, we develop a new model to tackle the CF problem which predicts user's ratings on previously unrated items by effectively exploiting interactions among review texts as well as the hidden user community and item group information. We call this model CMR (Co-clustering collaborative filtering Model with Review text). Specifically, we employ the co-clustering technique to model user communities and item groups. Each community–group pair corresponds to a co-cluster, which is characterized by a rating distribution in exponential family and a topic distribution. Given the community–group co-cluster of a certain entry in the user–item rating matrix, the corresponding observed rating score would be sampled from the rating distribution in exponential family of the given co-cluster. Likewise the corresponding review texts would be generated by a topic model which is governed by the same community–group co-cluster. Besides, most of the existing works [15,18,24] on co-clustering assume that each element of a variable belongs to only one cluster, which is too restrictive. For example, a user may be keen on football as well as basketball. Consequently, in our model, each user and item is modeled as a mixed membership over community and group, respectively, so each user or item can belong to multiple communities or groups with varying degrees. It should be noted that the number of hidden communities can differ with the number of hidden groups in our model to allow more flexible modeling. In order to evaluate our proposed CMR model, we have conducted extensive experiments on 22 real-world datasets, and our proposed model CMR outperforms the state-of-the-art latent factor models for the recommendation task. The results also clearly illustrate that considering interactions among review texts as well as co-clusters of hidden user communities and item groups provide valuable information to help improve the recommendation performance.

Furthermore, both the user preference and item profile are drifting over time. Specifically, it is quite common that users tend to shift their taste due to seasonal changes, holiday promotions, or even friends' recommendations from social networks [34]. Such changes in user's inclination could be reflected on user's rating behaviors, user's wording patterns in review comments as well as user's community affiliations. For example, if your best friend is crazy for the Korean drama and there are many TV promotions about Korean drama, you may easily follow the general trend and become a K-drama fan. Then you tend to buy Korean style clothings, and even comment with content about Korean. After a certain time, with Japanese dramas becoming more popular than Korean dramas, your taste may change. Similarly, an item profile may change with the continuous increase of review comments and rating scores. Consequently, dynamic modeling the temporal changes in user preference and item profile are desirable for improving a recommendation system [16,20,34,37,62,71]. In order to support the consideration of temporal aspect, the dataset for CF needs to be large and contains a wide time span. We extend our CMR model to consider time information and exploit the temporal interaction among review texts and co-clusters of user communities and item groups. The

المنارة للاستشارات

extended model is called TCMR (Time-aware Co-clustering collaborative filtering Model with Review text). In this TCMR model, each community–group co-cluster is characterized by an additional Beta distribution for time modeling. TCMR generates the observed rating scores and review texts by a similar community–group co-clustering approach as CMR. In addition, the Beta distribution in the co-cluster is employed to generate the observed reviewing time [58].

To evaluate our TCMR model, we have conducted another set of experiments on 22 larger datasets with wider time span. Our proposed model TCMR performs better than CMR and the standard time-aware recommendation model on the rating score prediction tasks. We also investigate the temporal effect on the user–item co-clusters.

The fundamental part of CMR has been published in a recent conference paper [64]. In this journal paper, we provide a more cohesive framework for presenting CMR. We also investigate the temporal CF problem and extend CMR to a new model called TCMR with the consideration of time information. We also conduct more extensive experiments and analyze the results to demonstrate the efficacy of our proposed models.

2 Related works

One of the most successful methods for collaborative filtering (CF) is latent factor model (LFM) which assumes that users' preference are determined by a small number of unobserved factors [47]. Low-rank matrix factorization [3,42,48,68] is a common implementation for the latent factor model. With the user–item rating matrix as input, the goal of low-rank matrix factorization is to infer the user matrix and item matrix, whose dot product operation can approximate the input matrix with minimal sum-squared loss. Then, the inferred user matrix and item matrix can be employed to predict the user's ratings on previously unrated items. These methods have been popular in recently years. Bell et al. [7] noticed that the neighborhood-based technique focuses more on patterns at local scale. Instead, the SVD-like matrix factorization technique performs better at higher, regional scale. Hence, they proposed an ensemble CF method by integrating these two complementary models to improve the performance. Koren [33,35] then proposed a more powerful CF algorithm known as SVD++ which is capable of exploiting both users' explicit and implicit feedbacks. Srebro and Jaakkola [52] developed a simple and efficient EM algorithm to approximate the target user–item rating matrix. Subsequently, they proved the generalization error bound of rating prediction [51]. Some other works [17,45,53,59,60] focus on low-rank matrix factorization based on the maximum margin principle. In addition, Salakhutdinov and Andriy [46,47] proposed probabilistic matrix factorization models to capture the uncertainty associated with each user–item rating.

Since simple LFM cannot easily be coupled with user–item interaction-associated information in different recommendation scenarios, researchers have recently explored to enhance the traditional LFM by exploiting rich features generated by users and items. One of the typical approaches is factorization machine (FM) [44] which is a class of model combining the advantage of support vector machine (SVM) and factorization models. In theory, FM makes it possible to incorporate any auxiliary information in user–item matrix. Hong et al. [32] extended FM to a model called cofactorization machines (CoFM) to handle multiple aspects of the dataset together with user's interest modeling. Qiang et al. [43] proposed a ranking factorization machine (ranking FM) model to consider various interaction features in microblog retrieval problem. Loni et al. [38] demonstrated the effectiveness of FM in the cross-domain CF. Geuens [25] developed a FM-based hybrid recommendation system by combining four

المنارة للاستشارات

different data sources including customer data, product data, implicit and explicit behavioral data. However, the existing formalism of FM has not been fully explored with topical modeling of the text content.

As the review texts are the main text source of user–item interaction, several existing works have incorporated review texts on recommendation tasks. Ganu et al. [22] harnessed the predictive ratings for recommendation benefiting from the manually discovered aspects of the reviewed item from review texts. Agarwal and Chen [1] proposed a matrix factorization method named fLDA model to improve the rating predictive accuracy, by taking advantage of the user's and item's review texts as regularizers. Another work proposed by Wang and Blei [57] recommends scientific articles to users based on the users' historic published articles and other users' ratings. Bao et al. [6] proposed a model simultaneously exploiting ratings and reviews for recommendation. Zhang et al. [70] developed an explainable recommendation system by extracting user opinions from the reviews at the phrase level. Diao et al. [19] proposed a probabilistic model jointly incorporating aspects, ratings, and sentiments for movie recommendation. Xin et al. [63] utilized review texts to improve the cross-domain collaborative filtering models. He et al. [29] devised a personalized explainable recommendation system by modeling the aspects discovered from the review texts. Guan et al. [27] proposed a phrase-based recommendation system with the consideration of review texts in the phrase level. McAuley and Leskovec [40] developed the HFT model for rating prediction. Essentially, HFT can be treated as a combination between traditional latent factor model and the topic model. The latent factor model is utilized to characterize each user and each item while the topic model is employed to model the corresponding review text. Then they apply the softmax function to connect the user or item factor vector and the topic distribution vector generated by the topic model. As a result, their predictive ratings can be adjusted to achieve less sum-squared loss by fitting the rating score as well as review text probabilistic likelihood. Our work differs from HFT in several aspects. Apart from incorporating the review text, our model is capable of explicitly exploiting the hidden community and group information embedded in the user and item collection, which is modeled by the co-clustering technique. Moreover, in HFT, the number of user factors is the same as the number of item factors, whereas in our approach, the number of user and item latent factors can be specified as different dimensions facilitating better modeling.

Besides, there are also some works concerning the user and item internal relationship for recommendation. Shan and Banerjee [49] treated the user–item rating matrix as a dyadic matrix, where each entry captures a relation between two entities of interest. They proposed a Bayesian co-clustering model, namely, simultaneously clustering of the users and items in the user–item rating matrix, to predict user's ratings on previously unrated items. Cai et al. [12] borrowed the ideas of object typicality from the cognitive psychology and proposed a typicality-based CF recommendation system which is capable of identifying the users' neighbors based on user typicality degrees in user community. Additionally, Beutel et al. [8] developed a unified Bayesian approach for CF. Based on the co-clustering of users and items, their model is able to automatically learn the most appropriate number of user clusters and item clusters by Chinese Restaurant Process. Chen et al. [13] devised an accurate and scalable recommendation system by weighting and ensembling the approximate matrices of multiple sets of the user–item co-clustering result. Heckel and Vlachos [30] proposed a recommendation algorithm by identifying overlapping co-clusters consisting of users and items. In contrast with our work, the above models ignore the valuable information from review texts.

Furthermore, the temporal effect has received some attentions recently. One focus is dynamic modeling of the user or item profiles. Chen et al. [14] developed a system that dis-

المنارة للاستشارات

covers, stores, and updates private dynamic user profiles for personalized recommendation. Chu and Park [16] proposed feature-based predictive bilinear regression models to provide accurate personalized recommendations of new items for both existing and new users. Baltrunas and Amatriain [5] proposed the micro-profiling approach for capturing the dynamic nature of user's taste. Another focus is to leverage the wisdom of crowds based on collaborative filtering (CF). Zimdars et al. [72] tracked the user's taste change by modeling CF as a univariate time series problem. Ding and Li [20] developed a time weighting scheme to assign decaying weights to previously rated items based on the time difference. Lathia et al. [37] formalized CF as a time-dependent, iterative prediction problem, and proposed a method to automatically assign and update per-user neighborhood sizes for parameter selection. Xiang et al. [62] extended the dynamic modeling to implicit feedback through random walk on Sessions-based Temporal Graphs (STG) for the user's short-term and long-term preference. Yu et al. [67] refined the STG model with the consideration of topic information. Their model has been deployed to learn user interest for tweet recommendations. Koren [34] proposed the CF-based time-aware algorithm timeSVD++. The timeSVD++ method assumes that the latent features composed of some time-evolving components and some others that are dedicated bias for each user at each specific time point. This method achieves an encouraging result on Netflix. Gailllard and Renders [21] proposed a time-sensitive collaborative filtering framework by means of adaptive matrix completion. Zhang et al. [71] took advantage of time series process to tackle with the year-long seasonal period of purchasing data to achieve daily-aware preference predictions, and then leverage the conditional opportunity models for daily-aware personalized recommendation with the help of explicit product features from the review texts. Unlike existing works, our models are flexible to consider three influential sources including rating scores, review texts, as well as time effect which are effectively modeled by the co-clustering technique.

3 Backgrounds

The problem we investigate in this paper slightly differs with the traditional collaborative filtering (CF) problem which only considers the rating score provided by a user for an item. Normally such rating information can be represented by a user–item rating matrix, which is typically partially filled. In addition to such rating information, we also take the review texts associated with each rating score as input. Formally, let $\mathcal{U} = \{1, 2, \ldots, U\}$ be the set of users and $\mathcal{V} = \{1, 2, \ldots, V\}$ be the set of items. Each user–item pair (u,v) corresponds to a rating score $r_{uv} \in \mathbb{R}_+$ and associates with a piece of review texts d_{uv}. The objective of this problem is to predict rating scores on the previously unrated items. Some major notations we use throughout the paper are defined in Table 1.

It is common that the rating score is discrete and ordinal (e.g., 1–5 stars). For each community–group co-cluster, we output a rating probabilistic distribution in exponential family. Such kind of rating distribution output has also been adopted in the work of Tan et al. [54], Beutel et al. [8], as well as Shan and Banerjee [49] because it can provide us more insightful information. In our case, it allows for each community–group co-cluster to output rating scores with varying degrees of rating uncertainty. Such rating distribution, instead of a single rating value, can allow better representation for the rating habit of each community–group co-cluster. We also output a topic distribution for each community–group co-cluster to uncover topics commonly discussed by a user community for an item group.

المنارة للاستشارات

Table 1 Notations

Symbol	Description
$\mathcal{U}$	User collection
$\mathcal{V}$	Item collection
r_{uv}, r_d	Rating score of the review d given by the user u for the item v
t_{uv}, t_d	Review time of the review d given by the user u for the item v
d_{uv}	Review text written by the user u for item v
w_{uvn}, w_{dn}	The nth word in the review text d_{uv}
z_{uvn}, z_{dn}	The topic for the nth word in the review text d_{uv}
π_u^c	K_1-dimensional community mixed membership vector for the user u
π_v^g	K_2-dimensional group mixed membership vector for the item v
c_{uv}, c_d	User community of d_{uv} and r_{uv}
g_{uv}, g_d	Item group of d_{uv} and r_{uv}
θ_{cg}, θ_{ij}	K-dimensional topic distribution for the co-cluster of the community–group pair (c, g) or (i, j)
ψ_{cg}, ψ_{ij}	S-dimensional rating distribution for the co-cluster of the community–group pair (c, g) or (i, j)
ξ_{cg}, ξ_{ij}	The parameter of the beta distribution for the co-cluster of the community–group pair (c, g) or (i, j)
ϕ_k	Word distribution for the topic k

4 Our proposed model: CMR

4.1 Model description

The graphical model of our proposed model is depicted in Fig. 1. Each node represents a variable. The shaded nodes represent the observed variables, and non-shaded nodes are the hidden variables to be inferred. The arrows indicate the dependency among the variables. The two outer rectangle plates represent the replication for a user and an item, respectively. The overlapping region indicates the rating scores and review texts associated with user–item pairs. The inner rectangle plate corresponds to each word in a review text.

As mentioned in the previous section, our proposed model CMR exploits the review texts and the hidden user community and item group information. Co-clustering technique is employed to effectively capture the relationship between users and items. In our CMR model, given K_1 community for users and K_2 groups for items, there are in total $K_1 \times K_2$ community–group co-clusters. The user $u \in \mathcal{U}$ is characterized by a K_1-dimensional community mixed membership vector π_u^c. Each item $v \in \mathcal{V}$ is similarly represented by a K_2-dimensional group mixed membership vector π_v^g group. Each vector component denotes the probability belonging to a certain user community or item group. For example, a sample π_u^c of $(0.5, 0.1, 0.4)$ indicates that the user u belongs to the community 1, 2, and 3 with the probability 0.5, 0.1, and 0.4, respectively. π_u^c, $u \in \mathcal{U}$ and π_v^g, $v \in \mathcal{V}$ are assumed to be generated by the Dirichlet prior distribution Dir(α) and Dir(β), respectively. Since Dirichlet function is the conjugate function of multinomial function, it is appropriate and computationally efficient to use Dirichlet distribution as the prior. For a user–item pair (u,v), i.e., an entry in the user–item rating matrix, the user community c_{uv} and the item group g_{uv},

المنارة للاستشارات

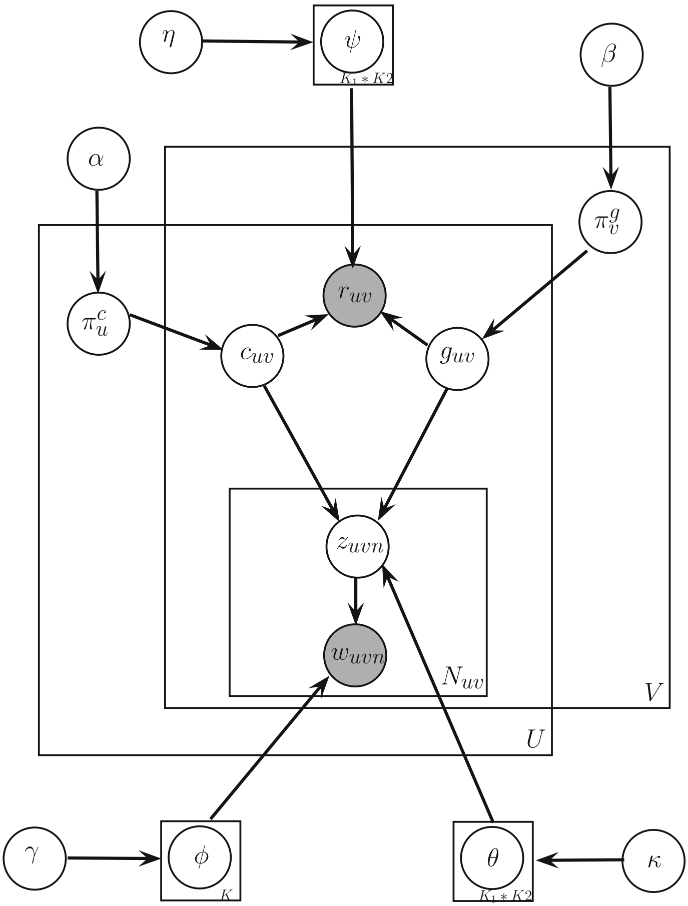


Fig. 1 Graphical representation of CMR. The *shaded and non-shaded nodes* represent the observed variables and the hidden variables to be inferred, respectively. The *two outer rectangle plates* represent the replication for a user and an item, respectively. The *overlapping region* indicates the rating scores and review texts associated with user–item pairs. The *inner rectangle plate* corresponds to each word in a review text. Specifically, a user community c_{uv} and an item group g_{uv} form a co-cluster which is modeled by the distributions θ and ψ for generating topics z and ratings r, respectively. The word w_{uvn} is generated from the topic z_{uvn}

which are sampled from the Multinomial distribution Multi(π_u^c) and Multi(π_v^g), respectively, would determine one co-cluster (c_{uv}, g_{uv}). This co-cluster is modeled by a rating distribution in exponential family $P(r_{uv}|\psi_{c_{uv}g_{uv}})$ and a topic distribution $\theta_{c_{uv}g_{uv}}$. Each user–item pair (u,v) is associated with an observed rating score and an observed review text. Regarding the generation of rating score, since the true rating score commonly takes on discrete integer (e.g. 1–5 stars) [54], $p(r_{uv}|\psi_{c_{uv}g_{uv}})$ can be simply modeled as Multinomial distribution Multi($\psi_{c_{uv}g_{uv}}$), where T-dimensional vector $\psi_{c_{uv}g_{uv}}$ denotes the probability distribution over each possible rating score. Hence, the predicted rating distribution for the given user u and the item v can be computed by Eq. 1 below,

$$P(r_{uv}|u, v) = \sum_{i,j} \pi_{ui}^c \pi_{vj}^g \psi_{ijr_{uv}} \tag{1}$$

المنارة للاستشارات

where i and j represent the index for the community and group, respectively. We also make use of a Dirichlet distribution $\mathrm{Dir}(\eta)$ as a prior on the rating distribution of each co-cluster. As a result, the observed rating score r_{uv} will be generated by $\mathrm{Multi}(\psi_{c_{uv}g_{uv}})$. On the other hand, the generation of each word in d_{uv} is modeled by the popular Latent Dirichlet Allocation (LDA) model [10] with the topic distribution $\theta_{c_{uv}g_{uv}}$ and the word distribution ϕ of K topics.

The generative process for all the rating scores and review texts is as follows:

- For each user $u \in \mathcal{U}$, choose $\pi_u^c \sim \mathrm{Dir}(\alpha)$
- For each item $v \in \mathcal{V}$, choose $\pi_v^g \sim \mathrm{Dir}(\beta)$
- For each co-cluster, choose topic distribution $\theta \sim \mathrm{Dir}(\kappa)$
- For each co-cluster, choose rating distribution $\psi \sim \mathrm{Dir}(\eta)$
- For each topic, choose word distribution $\phi \sim \mathrm{Dir}(\gamma)$
- For each user–item pair (u,v)
    - Choose $c_{uv} \sim \mathrm{Multi}(\pi_u^c)$, $g_{uv} \sim \mathrm{Multi}(\pi_v^g)$
    - Choose rating score $r_{uv} \sim \mathrm{Multi}(\psi_{c_{uv}g_{uv}})$
    - For each word n in associated review text d_{uv}
        - Choose the topic $z_{uvn} \sim \mathrm{Multi}(\theta_{c_{uv}g_{uv}})$
        - Choose word w_{uvn} from the word distribution $\phi_{z_{uvn}}$

In contrast with the traditional latent factor models such as Salakhutdinov and Andriy [46, 47], McAuley and Leskovec [40], Bell et al. [7] and Koren [33], our model is capable of capturing more realistic situations. Particularly, the user community's rating behavior and the commonly reviewed topics can vary with different item groups. For instance, a user community, who is keen on sports, tends to give high ratings for the product items of some famous brands such as Nike and Adidas while it gives relatively low ratings for those unknown brands. On the other hand, regarding a certain T-shirt group, terms such as "comfortable," "durable," and "good quality" would frequently appear in the reviews written by the user community, who cares more about the quality and the comfort of clothing, whereas the user community concerning the style and appearance of T-shirts would provide comments including key terms such as "beautiful," "nice," and "fashion." Many existing models represent each user and item with a hidden factor vector for handling this situation, but they cannot effectively model the nature of user communities and item groups. In contrast, our CMR model can provide a more precise modeling for these concepts. Besides, in traditional latent factor models, the feature vector for each user and each item has typically the same dimension. In other words, the number of factors for characterizing each user is the same with that of each item. However, the number of user modeling factors should differ from the number of item modeling factors. The reason is that the hidden community partitions in the users would be more complicated and can vary with different influencing factors. For example, if the clothings are assumed to form three groups including T-shirt, pants, and underwear, due to the interest overlapping in different communities, the users can be clustered into more than three communities—a community who tends to buy T-shirt, a community who tends to buy pants, a community who tends to buy T-shirt and pants, and a community who tends to buy pants and underwear, etc. Additionally, some communities cannot be represented by common group partition of items. If a user community is the fans of David Beckham, they would like to buy the clothings endorsed by Beckham. However, in common e-commerce sites, there is usually no specific item category related to Beckham endorsed clothings. Consequently, discriminating the modeling of user latent factor and item latent factor is a better approach.

المنارة للاستشارات

4.2 Posterior inference

Exact inference for the CMR model is intractable. We employ the collapsed Gibbs sampling algorithm to perform approximate inference. Gibbs sampling is a special case of Metropolis–Hastings algorithm which is one of Markov Chain Monte Carlo methods [23]. It is used to obtain a sample approximated from a joint distribution when only the conditional distributions of each variable can be efficiently computed [55,56]. Variables considered in Gibbs sampling are sequentially sampled from the distribution conditioned on all the other variables. Due to the Markov property, the chain of the model states would converge to a stationary sample from the joint distribution. Typically, when Gibbs sampling is employed to do inference for LDA, Griffiths and Steyvers [26] proposed a collapsed Gibbs sampling method to demonstrate that we just need to sample the topic assignments $\mathbf{z}$ in that the dependency on topic distribution θ and word distribution φ can be analytically integrated out. Therefore, for our CMR model, we only sample the assignments of user communities $\mathbf{c}$, item groups $\mathbf{g}$, and the topics $\mathbf{z}$ by integrating out the community mixed membership vectors $\boldsymbol{\pi}^{\mathbf{c}}$, the group mixed membership vectors $\boldsymbol{\pi}^{\mathbf{g}}$, the rating distribution $\boldsymbol{\psi}$, the topic distribution $\boldsymbol{\theta}$, and the word distribution $\boldsymbol{\phi}$.

For Gibbs sampling with our CMR model, we need to compute the conditional distribution below,

$$P(z_{uvn}=k, c_{uv}=i, g_{uv}=j|\mathbf{z}^{\neg uvn}, \mathbf{c}^{\neg uvn}, \mathbf{g}^{\neg uvn}, \mathbf{r}, \mathbf{w}) \tag{2}$$

where $\mathbf{z}^{\neg uvn}$, $\mathbf{c}^{\neg uvn}$, $\mathbf{g}^{\neg uvn}$ are vectors of the topic assignments, the user community assignments, and the item group assignments, respectively, without considering the nth word of the review text written by the user u for the item v. $\mathbf{r}$ and $\mathbf{w}$ are the vectors representing all the rating scores and all the words in associated review texts, respectively. i, j, and k are the assignment of user community, item group, and topic, respectively, for the current considered word. We begin with the joint distribution of our model and collapse out all the intermediate latent variables including π^c, π^g, ψ, θ, and ϕ. It should be noted that similar with the time stamp shared by all the words in the documents in the TOT (Topic Over Time) model [58], each word of d_{uv} in our model can have the same community–group co-cluster (i, j) and rating score. Once updating the topic assignment of each word, the corresponding community and group should also be updated synchronously. Given that the considered nth word in the review text d_{uv} is denoted by x, and the associated rating score is represented by s, we present the final conditional probability in Eq. 2 with the help of the chain rule as follows.

$$\begin{aligned}
&P(z_{uvn}=k, c_{uv}=i, g_{uv}=j|\mathbf{z}^{\neg uvn}, \mathbf{c}^{\neg uvn}, \mathbf{g}^{\neg uvn}, \mathbf{r}, \mathbf{w}) \\
&\quad\propto \left(\alpha_i + e_{u,(\cdot),i,(\cdot)}^{(\cdot),(\cdot),(\cdot),\neg uvn}\right) \times \left(\beta_j + e_{(\cdot),v,(\cdot),j}^{(\cdot),(\cdot),(\cdot),\neg uvn}\right) \\
&\quad\times \frac{\eta_s + e_{(\cdot),(\cdot),i,j}^{s,(\cdot),(\cdot),\neg uvn}}{\sum_s \left(\eta_s + e_{(\cdot),(\cdot),i,j}^{s,(\cdot),(\cdot),\neg uvn}\right)} \times \frac{\kappa_k + e_{(\cdot),(\cdot),i,j}^{(\cdot),k,(\cdot),\neg uvn}}{\sum_k \left(\kappa_k + e_{(\cdot),(\cdot),i,j}^{(\cdot),k,(\cdot),\neg uvn}\right)} \\
&\quad\times \frac{\gamma_x + e_{(\cdot),(\cdot),(\cdot),(\cdot)}^{(\cdot),k,x,\neg uvn}}{\sum_x \left(\gamma_x + e_{(\cdot),(\cdot),(\cdot),(\cdot)}^{(\cdot),k,x,\neg uvn}\right)}
\end{aligned} \tag{3}$$

Here, we have introduced a major notion e for word counting. $e_{u,v,i,j}^{s,k,x,\neg uvn}$ indicates the number of words whose topic assignment is k and has word index of x, and that appear

المنارة للاستشارات

in the review text d_{uv} with the rating score of s and belongs to the co-cluster (i, j). There are in total 7 dimensions, and any $(\cdot)$ operator represents the counting of the words without considering the corresponding dimension. Assume that the nth word of review text d_{uv} is excluded, $e^{(\cdot),(\cdot),(\cdot),\neg uvn}_{u,(\cdot),i,(\cdot)}$ is the number of words written by the user u and attached with the user community i. Similarly, $e^{(\cdot),(\cdot),(\cdot),\neg uvn}_{(\cdot),v,(\cdot),j}$ denotes the number of words for describing the item v and attached with the item group j. Besides, $e^{s,(\cdot),(\cdot),\neg uvn}_{(\cdot),(\cdot),i,j}$ captures the number of words generated by the co-cluster (i,j) and attached with the rating s while $e^{(\cdot),k,(\cdot),\neg uvn}_{(\cdot),(\cdot),i,j}$ also indicates the number of words generated by the co-cluster (i,j) but requiring the topic assignment of k. The last counter $e^{(\cdot),k,x,\neg uvn}_{(\cdot),(\cdot),(\cdot),(\cdot)}$ represents the number of words whose topic assignment is k and the word index is x.

The community mixed membership vector for the user u and the group mixed membership vector for the item v can be estimated by a sample of such Markov chain as follows,

$$\pi^{c}_{ui} = \frac{\alpha_i + e^{(\cdot),(\cdot),(\cdot)}_{u,(\cdot),i,(\cdot)}}{\sum_i \left(\alpha_i + e^{(\cdot),(\cdot),(\cdot)}_{u,(\cdot),i,(\cdot)}\right)} \tag{4}$$

$$\pi^{g}_{vj} = \frac{\beta_j + e^{(\cdot),(\cdot),(\cdot)}_{(\cdot),v,(\cdot),j}}{\sum_j \left(\beta_j + e^{(\cdot),(\cdot),(\cdot)}_{(\cdot),v,(\cdot),j}\right)} \tag{5}$$

Moreover, the posterior estimates for the rating distribution ψ, the topic distribution θ, and the word distribution ϕ can be computed below,

$$\psi_{ijs} = \frac{\eta_s + e^{s,(\cdot),(\cdot)}_{(\cdot),(\cdot),i,j}}{\sum_s \left(\eta_s + e^{s,(\cdot),(\cdot)}_{(\cdot),(\cdot),i,j}\right)} \tag{6}$$

$$\theta_{ijk} = \frac{\kappa_k + e^{(\cdot),k,(\cdot)}_{(\cdot),(\cdot),i,j}}{\sum_k \left(\kappa_k + e^{(\cdot),k,(\cdot)}_{(\cdot),(\cdot),i,j}\right)} \tag{7}$$

$$\phi_{kx} = \frac{\gamma_x + e^{(\cdot),k,x}_{(\cdot),(\cdot),(\cdot),(\cdot)}}{\sum_x \left(\gamma_x + e^{(\cdot),k,x}_{(\cdot),(\cdot),(\cdot),(\cdot)}\right)} \tag{8}$$

The Gibbs sampling procedure can be performed using Eqs. 3 to 8. After a random initializing, during the process of Gibbs sampling, we take an interval of L iterations between subsequent read-outs to obtain a steady approximate solution [26]. The detailed algorithm is described in Algorithm 1.

5 Experiment on CMR model

We demonstrate the effectiveness of our model by conducting extensive experiments on 22 real-world datasets covering different product categories. We also compare with a basic linear method (SVM) and the state-of-the-art approaches.

المنارة للاستشارات

Algorithm 1 Gibbs sampling for CMR model

```
Input: A collection of rating scores $r_{uv}$ given by the users $u \in \mathcal{U}$ for the items $v \in \mathcal{V}$. Each rating score $r_{uv}$
  is associated with a review text $d_{uv}$.
Output: Latent variables $\pi_u^c, \pi_v^g, \psi, \theta, \phi, c_{uv}, g_{uv}, z_{uvn}$
  Initialize $c_{uv}, g_{uv}, z_{uvn}$ randomly for all the words
  for $iter = 1 \rightarrow Max_{iter}$ do
    for all $d_{uv}, u \in \mathcal{U}, v \in \mathcal{V}$ do
      for all $w_{uvn} \in d_{uv}$ do
        draw $z_{uvn}, c_{uv}, g_{uv}$ from Eq. 2
        update word counters $e_{u,(\cdot),i,(\cdot)}^{(\cdot),(\cdot),(\cdot)}, e_{(\cdot),v,(\cdot),j}^{(\cdot),(\cdot),(\cdot)}, e_{(\cdot),(\cdot),i,j}^{s,(\cdot),(\cdot)}, e_{(\cdot),(\cdot),i,j}^{(\cdot),k,(\cdot)}, e_{(\cdot),(\cdot),(\cdot),(\cdot)}^{(\cdot),k,x}$
      end for
    end for
    if converged or $iter\%L = 0$ then
      read out $\pi_u^c, \pi_v^g, \psi, \theta, \phi$ by Eqs. 4 to 8
    end if
  end for
```

5.1 Datasets

We use 22 datasets[1] crawled from Amazon[2] in a wide range of product categories. Such datasets have also been utilized in McAuley et al. [41], McAuley and Leskovec [39,40]. In particular, each dataset is a collection of review comments from a set of users for the product items, and each review text is accompanied with a rating score (e.g., 1–5 stars) to show the user's overall satisfactory level for the reviewed product item. Note that we randomly sample a portion of some very large datasets (e.g., over GB) by limiting the number of items up to 5000 in a similar manner as in [6]. A detailed summary of the entire datasets is reported in Table 2.

5.2 Comparative methods and evaluation metric

The HFT proposed by McAuley and Leskovec [40] demonstrates the state-of-the-art performance to predict the user's rating by exploiting the review texts. Thus, we compare with this method. Note that, there are two versions of HFT. HFT(item) associates each item latent factor with the topics expressed in the review texts related to each item while HFT(user) conducts similar procedure for each user. Since HFT(item) has been shown better performance than HFT(user) [40], we compare our CMR model with HFT(item) in our experiment. Moreover, we conduct comparison with a basic SVM method [69] and the state-of-the-art probabilistic latent factor methods including probabilistic matrix factorization (PMF) and its Bayesian version (BPMF). However, these three methods just consider the rating information. For SVM, we train a multi-class SVM classifier with probabilistic output for each user. All the training and test samples for a particular SVM are items rated by a particular user, and the rating values are represented as class labels.

Beutel et al. [8] pointed out that minimizing the mean square loss is a mainstay in CF, but there are a number of other better metrics. For example, Weimer et al. [59] and Yang et al. [65] treated CF as a preference ranking problem rather than directly predict the rating score. Distance similarity [61] is also utilized to evaluate the performance of CF. Besides, in order to consider the uncertainty and discrete characteristic (e.g., 1–5 stars) of the rating score, Tan et al. [54], Beutel et al. [8], and Shan and Banerjee [49] introduced the discrete

[1] http://snap.stanford.edu/data/web-Amazon.html.

[2] www.amazon.com.

المنارة للاستشارات

Table 2 Dataset statistics for CMR model

Dataset	No of. users	No of. items	No of. reviews	Avg. words	Time span
Amazon Instant Video	97,152	5000	160,120	71.01	Feb. 2002–Aug. 2008
Arts	24,071	4211	27,980	37.92	Apr. 1998–Mar. 2013
Automotive	133,256	47,577	188,728	38.75	Oct. 1998–Mar. 2013
Baby	123,837	6962	184,887	45.63	Feb. 1999–Mar. 2013
Beauty	167,725	29,004	252,056	37.85	Jan. 1997–Mar. 2013
Cell Phones Accessories	68,041	7438	78,930	50.05	Nov. 1999–Mar. 2013
Clothings	15,782	5000	50,127	32.56	Mar. 2004–Mar. 2008
Electronics	73,032	5000	83,094	52.36	Feb. 2002–Jul. 2008
Gourmet Foods	112,544	23,476	154,635	38.94	Jun. 1998–Mar. 2013
Health	46,416	5000	55,201	40.48	Dec. 2005–Oct. 2009
Industrial Scientific	29,590	22,622	137,042	31.21	Aug. 1998–Mar. 2013
Jewelry	40,594	18,794	58,621	29.90	Feb. 1999–Mar. 2013
Kindle Store	116,191	4372	160,793	73.48	Jul. 1995–Mar. 2013
Musical Instruments	67,007	14,182	85,405	46.69	Apr. 1998–Mar. 2013
Office Products	110,472	14,224	138,084	43.57	Jun. 1997–Mar. 2013
Patio	166,832	19,531	206,250	44.08	Nov. 1998–Mar. 2013
Pet Supplies	160,496	17,523	217,170	43.37	Apr. 2000–Mar. 2013
Shoes	12,045	5000	33,462	35.54	Mar. 2001–Mar. 2006
Software	68,464	11,234	95,084	63.02	Nov. 1997–Mar. 2013
Tools Home Improvement	38,123	5000	42,154	43.95	Nov. 2005–Oct. 2009
Toys Games	38,328	5000	46,232	40.48	Feb. 2002–Jun. 2007
Watches	62,041	10,318	68,356	42.73	Dec. 1998–Mar. 2013

rating probabilistic distribution and employed the (negative) log-likelihood to measure the fit of their model. Similarly, we employ the negative rating log-likelihood called NLL in Eq. 9 as a metric to measure the model's fit for the observed rating. Smaller NLL indicates a better rating prediction performance.

$$\mathrm{NLL} = \frac{1}{N} \sum_{u,v} - \log P(r_{uv} | u, v) \tag{9}$$

The NLL metric requires the computation of $P(r_{uv}|u, v)$. For our CMR model, we implement the general exponential distribution family by simple multinomial distribution for modeling the rating score and compute $P(r_{uv}|u, v)$ by Eq. 1. For SVM, the trained model can estimate the probability of the class label, i.e., the rating r_{uv}. The NLL metric for SVM can be calculated as in Eq. 9.

For other comparative models, the latent factor model can be alternatively treated as a Gaussian noise model [54]. The observed rating r_{uv} is estimated by a Gaussian distribution with the mean of the predictive rating $\hat{r}_{uv}$. Therefore, we obtain $r_{uv} \sim N(\hat{r}_{uv}, \tau^{-1})$, i.e.,

المنارة للاستشارات

$$-\log P(r_{uv}|u,v)=\frac{\tau}{2}(r_{uv}-\hat{r}_{uv})^2+\frac{1}{2}\log 2\pi-\frac{1}{2}\log\tau \tag{10}$$

where τ is the precision of the Gaussian distribution, and the specific form of predictive rating $\hat{r}_{uv}$ varies with different latent factor models. However, it is common that the real rating score r_{uv} is a discrete integer. Hence, it is more appropriate to model the observed ratings by normalized exponential family model. Given the rating score belongs to the common 1–5 stars, i.e., $\mathcal{Y}=\{1,2,\ldots,5\}$, the corresponding Gaussian distribution can be reformulated as in Eq. 11.

$$P(r_{uv}|u,v)=\frac{e^{-\frac{\tau}{2}(r_{uv}-\hat{r}_{uv})^2}}{\sum_{q\in\mathcal{Y}}e^{-\frac{\tau}{2}(q-\hat{r}_{uv})^2}} \tag{11}$$

Consequently, Eq. 10 can be replaced by Eq. 12 to compute the NLL metric for HFT, PMF, and BPMF.

$$-\log P(r_{uv}|u,v)=\frac{\tau}{2}(r_{uv}-\hat{r}_{uv})^2+\log\sum_{q\in\mathcal{Y}}e^{-\frac{\tau}{2}(q-\hat{r}_{uv})^2} \tag{12}$$

5.3 Experimental setup

We conduct preprocessing on the datasets by removing punctuations, stop words from a standard stop word list as in Lacoste-Julien et al. [36] and converting the words into lower cases. In our experiment, the number of maximum iterations of the Gibbs sampler is set to 1000, and the inferred latent variables are computed every 100 times. In other words, $Max_{iter}=1000$ and $L=100$ in Algorithm 1. In order to perform fair comparison with HFT, for each dataset in Table 2, we randomly select 80% of the reviews to form the training set by limiting the maximum reviews up to 2 millions and then uniformly divide the remaining part into validation set and test set. Each component of hyperparameters is specified equally by the following values: $\alpha_l=\frac{0.5}{K_1}$, $\beta_m=\frac{0.5}{K_2}$, $\eta_t=\frac{0.5}{T}$, $\kappa_k=\frac{0.5}{K}$, and $\gamma_s=0.01$. The setting for K_1, K_2, and K will be described below.

5.4 Results on rating prediction

In our model, we fix the number of topics K as 5, which is similarly done in McAuley and Leskovec [40] and perform the grid search for the number of user community K_1 and item group K_2 in the range of [1, 15] and [1, 10], respectively, using the validation set. For each dataset, we choose K_1 and K_2 with the lowest NLL on the validation set and then calculate the corresponding NLL on the test set. For the comparative methods, we use the parameter setting in their papers [40,46,47], and also calculate the NLL metric on the test set.

The results in terms of NLL are reported in Table 3, where the best performance is bolded. In general our CMR model achieves the best performance on most datasets. Significantly best improvement can be achieved for "Clothings" and "Shoes." These results can reflect that it is more evident for the existence of user communities and item groups in these domains. Users would subjectively reveal their feelings or attitude toward the items they reviewed in their own way. Besides, there are also apparent item groups for "Clothings" and "Shoes." As a result, by capturing these hidden user communities and item groups, our CMR model can improve the rating prediction performance significantly. As for the basic SVM model, the performance is not stable. Although there are two datasets for which the results of SVM are good, SVM is not comparable to HFT or CMR in most datasets. Moreover, it can be observed that our CMR model and HFT perform better than that of PMF and BPMF demonstrating the effectiveness

المنارة للاستشارات

Table 3 Rating prediction performance

Method	Arts	Automotive	Baby	Beauty	Cell phones accessories	Clothings	Pet supplies	Gourmet foods	Health	Kindle store	Industrial and scientific
SVM	1.599	1.497	1.575	1.426	**1.433**	1.570	1.422	1.600	1.494	1.530	1.268
PMF	1.497	1.530	1.777	1.523	1.755	0.945	1.576	1.493	1.723	1.602	1.070
BPMF	1.484	1.519	1.598	1.473	1.670	0.879	1.555	1.486	1.635	1.557	0.916
HFT	1.398	1.410	1.426	1.374	1.656	0.823	1.458	1.414	1.436	1.440	0.787
CMR	**1.288**	**1.277**	**1.222**	**1.142**	1.493	**0.334**	**1.411**	**1.104**	**1.414**	**1.323**	**0.597**
Method	Jewelry	Musical instrument	Office products	Patio	Shoes	Software	Amazon instant video	Tool and home	Toys	Watches	Electronics
SVM	1.538	1.510	1.488	1.462	1.415	**1.430**	1.553	1.362	1.332	1.545	1.542
PMF	1.440	1.506	1.618	1.623	1.071	1.776	1.534	1.697	1.539	1.504	1.751
BPMF	1.423	1.501	1.595	1.614	0.913	1.697	1.507	1.603	1.476	1.489	1.649
HFT	1.330	1.415	1.488	1.514	0.800	1.650	1.438	1.485	1.402	1.453	1.532
CMR	**1.138**	**1.183**	**1.313**	**1.287**	**0.345**	1.462	**1.378**	**1.241**	**1.229**	**1.209**	**1.420**

Lower values indicate better results

المنارة للاستشارات

Table 4 Statistical significance tests for CMR and related models

Method	Average NLL
SVM	$1.481^{\ddagger}$
PMF	1.525
BPMF	$1.465^{\ddagger}$
HFT	$1.369^{\dagger\ddagger\S}$
CMR	$\mathbf{1.173}^{\dagger\ddagger\S\P}$

†, ‡, §, ¶ indicate that it is statistical significant at the significance level of 0.05 over SVM, PMF, BPMF, and HFT, respectively

of considering review texts. We also conduct the t test for the results, with the significance level being 0.05. The significance test results are shown in Table 4. It is statistical significant that CMR is better than all other models. Besides CMR, it is also statistical significant that HFT is a better method than other models.

5.5 Qualitative analysis of co-clusters

Table 5 exhibits the discovered topics for our CMR model, which is quite indicative to be interpreted and easy to discriminate. In Table 5a, CMR discovers 5 topics for the Clothings dataset and those topics can be interpreted as "Bags," "Pants," "Service," "Vest," and "Bras." For the Shoes dataset, CMR extracts 5 topics which are interpreted as "Service," "Leather Shoes," "Size," "Appearance," and "Boots" in Table 5b.

In our CMR model, a topic distribution θ_{cg} and a rating distribution ψ_{cg} will be inferred for each community–group co-cluster (c, g), which provides valuable information for discovering the commonly reviewed topics and rating habit for user communities and item groups. With the most appropriate $K_1 = 11$ and $K_2 = 6$ on the Shoes dataset, Fig. 2 depicts the topic distribution (TD) and rating distribution (RD) of different co-clusters. First, we fix the item group to be 6 with different user communities, namely, 1, 2, 6, and 9. As we can see in Fig. 2a, b, different user communities concern different aspects of the item group and have different rating habits. For example, depending on the topics shown in Table 5b, the users of $c = 2$ usually comment on the "Appearance" of the "Boots" and tend to give low ratings on this item. On the other hand, the users of $c = 6$ and $c = 9$ focus more on the "Size" and "Service" of the "Boots," and feel satisfied by giving high ratings. Second, we fix the user community to be 4 with varying item groups 1, 2, 5, and 6. Figure 2c, d illustrates that when dealing with different item groups, a certain user community also tends to vary its wordings and expression patterns as well as the rating habit. Specifically, for the item group corresponding to $g = 1$, the considered user community comments more about its "Size" and tends to give a descent rating, while this user community writes more about the leather shoes and boots in the reviews and usually gives a relatively high rating for the item group corresponding to $g = 5$.

6 Extension for time-awareness: TCMR

6.1 Model description

We extend the proposed CMR model to TCMR by incorporating the temporal aspect. The graphical model of our proposed TCMR model is depicted in Fig. 3, and the major extension

المنارة للاستشارات

Table 5 Top twenty words from each topic of Clothings and Shoes datasets for our CMR model

Bags	Pants	Service	Vest	Bras
(a) Clothings				
Bag	Jeans	Fit	Vest	Bra
Backpack	Fit	Size	Jacket	Size
Perfect	Pants	Bought	Color	Comfortable
Recommend	Pair	Pants	Pockets	Fit
Shirt	Size	Received	Price	Bras
Comfortable	Quality	Return	Warm	Wear
Wear	Levi	Amazon	Pocket	Support
Bought	Price	Wear	Quality	Cup
Time	Amazon	Disappointed	Fabric	Shape
School	Cut	Larger	Thin	Straps
Son	Carhartt	Slacks	Columbia	Love
Cool	Wear	Purchase	Light	Top
Durable	Waist	Shipping	Weight	Cotton
Love	Looking	Dockers	Fleece	Look
Colors	Socks	Shirt	Loops	Tight
Quality	Belt	Time	Short	Day
Soft	Bought	Front	Zipper	Cheap
Easy	Boot	Quality	Looking	Looks
Hold	Levis	Company	Leg	Pretty
Price	Brand	Waist	Bottom	Worn
(B) Shoes				
Pair	Pair	Shoes	Comfortable	Boot
Time	Foot	Fit	Shoe	Feet
Money	Toe	Wear	Love	Pair
Purchase	Pretty	Wide	Style	Comfortable
Buy	Sole	Bought	Look	Wear
Sneakers	Hard	Quality	Pair	Hiking
Months	Pairs	Feet	Wear	Break
Received	Stars	Narrow	Price	Support
Sandals	Black	Return	Perfect	Price
Shipping	Arch	Look	Bought	Warm
Store	Brown	Heel	Looking	Walking
Arrived	Sore	True	Colors	Time
Worth	Top	Slippers	Recommend	Waterproof
Service	Shoes	Foot	Happy	Recommend
Online	Rubber	Half	Feet	Worn
Brand	Soft	Leather	Running	Light
Days	Wider	Width	Walking	Fit
Purchased	Satisfied	Tight	White	Socks
Keds	Worn	Sole	Worn	Wet

المنارة للاستشارات

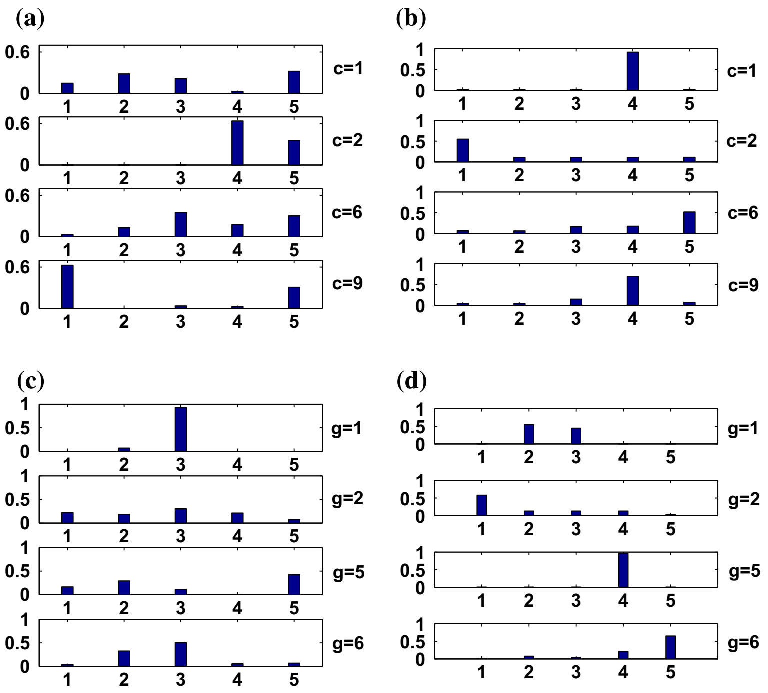


Fig. 2 Topic distribution (TD) and rating distribution (RD) of different community–group co-clusters on Shoes dataset. **a**, **b** are for fixed item group with varying user communities shown in the *right*. **c**, **d** are for fixed user community with varying item groups shown in the *right*. **a** TD for $g = 6$. **b** RD for $g = 6$. **c** TD for $c = 4$. **d** RD for $c = 4$

is the additional incorporation of the time information. In TCMR, for a user–item pair (u, v), its corresponding co-cluster (c_{uv}, g_{uv}) is characterized by a rating distribution in exponential family $P(r_{uv}|\psi_{c_{uv}g_{uv}})$, a topic distribution $\theta_{c_{uv}g_{uv}}$ as well as an additional beta distribution for time modeling. Each user–item pair (u, v) is associated with an observed rating score r_{uv}, the observed review text d_{uv}, and an observed time t_{uv}. Similarly, TCMR generates the observed rating score by Multinomial distribution Multi($\psi_{c_{uv}g_{uv}}$) and generates the observed review text with the topic distribution $\theta_{c_{uv}g_{uv}}$ and the word distribution ϕ of K topics. We employ the stochastic variational inference to estimate all the hidden variables and hence compute the rating distribution for a given user–item pair (u, v) by Eq. 13 below,

$$P(r_{uv}|u, v) = \sum_{i,j} \widehat{\sigma_{ui}^{c}} \widehat{\sigma_{vj}^{g}} m_{ijr_{uv}} \tag{13}$$

where $\widehat{\sigma_u^c}$ and $\widehat{\sigma_v^g}$ are the normalized form of the user community mixture membership vector σ_u^c and the item group mixture membership vector σ_v^g, namely, the variational parameters of π_u^c and π_v^g defined in the graphical model of TCMR, respectively.

TCMR first normalizes the time to the range of [0,1] by $t_n = \frac{t - t_{min}}{t_{max} - t_{min}}$ and then takes the temporal effect into consideration. The normalized time t_{uv} is generated by the beta distribution Beta($\xi_{c_{uv}g_{uv}}$) of co-cluster (c_{uv}, g_{uv}). In contrast with some existing works incorporating time [58], our TCMR model takes advantage of the underlying user–item relationship and employs the co-cluster to generate the observed time. Besides, TCMR further considers the

المنارة للاستشارات

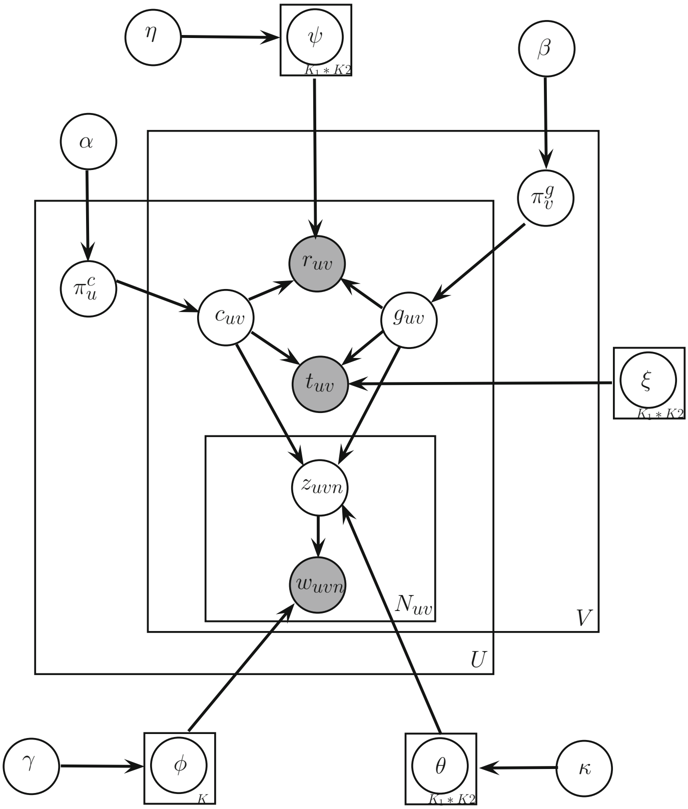


Fig. 3 Graphical representation of TCMR. The *shaded and non-shaded nodes* represent the observed variables and the hidden variables to be inferred, respectively. The *two outer rectangle plates* represent the replication for a user and an item, respectively. The *overlapping region* indicates the rating scores and review texts associated with user–item pairs. The *inner rectangle plate* corresponds to each word in a review text. Compared to CMR, for each co-cluster $c_{uv}g_{uv}$ in TCMR, there is an additional distribution ξ for generating the observed time t

valuable review texts which is ignored by most of the dynamic user modeling algorithms [34,66]. The entire generative process for all the rating scores, review texts, and reviewing times is as follows:

- For each user $u \in \mathcal{U}$, choose $\pi_u^c \sim \text{Dir}(\alpha)$
- For each item $v \in \mathcal{V}$, choose $\pi_v^g \sim \text{Dir}(\beta)$
- For each co-cluster, choose topic distribution $\theta \sim \text{Dir}(\kappa)$
- For each co-cluster, choose rating distribution $\psi \sim \text{Dir}(\eta)$
- For each topic, choose word distribution $\phi \sim \text{Dir}(\gamma)$
- For each user–item pair (u,v)
    - Choose $c_{uv} \sim \text{Multi}(\pi_u^c)$, $g_{uv} \sim \text{Multi}(\pi_v^g)$
    - Choose rating score $r_{uv} \sim \text{Multi}(\psi_{c_{uv}g_{uv}})$
    - Choose reviewing time $t_{uv} \sim \text{Beta}(\xi_{c_{uv}g_{uv}})$
    - For each word n in associated review text d_{uv}
        - Choose topic $z_{uvn} \sim \text{Multi}(\theta_{c_{uv}g_{uv}})$
        - Choose word w_{uvn} from word distribution $\phi_{z_{uvn}}$

المنارة للاستشارات

Similar to the CMR model, TCMR inherits its major advantages in co-clustering modeling for the rating score and the review text. The co-clustering technique in TCMR is capable of capturing the user's dynamic rating behavior and the commonly discussed topics for different item groups. Besides, in TCMR we model each reviewing time by a beta distribution of a community–group co-cluster since a user community's reviewing time tends to vary with different item groups. For example, most users would buy T-shirts instead of cotton-padded jackets in the summer, while some mountaineering fans or expeditions still need cold protective clothing even in the summer. On the other hand, since a user's probability of belonging to a certain co-cluster is changing over the time, TCMR can determine a more accurate co-cluster probability distribution for a user in a given time, and hence the rating prediction performance can also be improved.

We further analyze the rationale of the time effect. For a particular user u, his corresponding community mixture membership vector σ_u^c is positive correlated with the community mixture membership vector of all of his written review texts $\lambda_d^c, d \in D_u$. Given the fixed Beta parameter set ξ, we can find that ξ_i plays one of the major roles for determining the user u's probability belonging to the community i. Essentially, if $\xi_{i0}, \xi_{i1}, \ldots, \xi_{iK_2}$ can better fit the time t_d with higher likelihood, we have a larger λ_{di}^c leading to user u's larger probability belonging to the community i. In other words, considering time can help cluster the users who provide reviews in close time frame into the same user community. Therefore, the predicted rating performance can benefit from a more accurate co-cluster distribution.

6.2 Posterior inference

Given the model parameters $\Phi = (\alpha, \beta, \eta, \gamma, \kappa, \xi)$, the key procedure of applying the TCMR model is to infer the posterior distribution of the hidden variables set $\Theta = (\boldsymbol{\pi}^{\mathbf{c}}, \boldsymbol{\pi}^{\mathbf{g}}, \mathbf{c}, \mathbf{g}, \mathbf{z}, \boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\theta})$ conditioned on the observed rating score $\mathbf{r}$, the review text $\mathbf{w}$, and the time $\mathbf{t}$. Since TCMR is designed for large datasets with wide time span, we resort to the stochastic variational inference (SVI) method to infer the hidden variables due to its computational efficiency. The main idea behind SVI is to optimize the free parameters of a variational distribution over the hidden variables so that the approximated variational distribution is close to the true posterior with the minimal Kullback–Leibler divergence. Differing with the traditional batch variational inference method, SVI is capable of updating the global variational parameters by only scanning a small set of observations (Minibatch learning) with the help of online stochastic optimization, so that SVI can handily analyze massive document collections including those documents arriving in a stream [31].

Specifically, we introduce a family of factorized variational distribution for the hidden variables Θ. Given the free variational parameters $\Delta = (\boldsymbol{\sigma}^{\mathbf{c}}, \boldsymbol{\sigma}^{\mathbf{g}}, \boldsymbol{\lambda}^{\mathbf{c}}, \boldsymbol{\lambda}^{\mathbf{g}}, \boldsymbol{\mu}, \mathbf{m}, \mathbf{l}, \mathbf{h})$, we have

$$q(\Theta|\Delta) = \prod_u q(\pi_u^c|\sigma_u^c) \prod_v q(\pi_v^g|\sigma_v^g) \prod_{c,g} q(\psi_{cg}|m_{cg}) q(\theta_{cg}|h_{cg}) \prod_k q(\phi_k|l_k) \prod_d q(c_d|\lambda_d^c) q(g_d|\lambda_d^g) \prod_{d,n} q(z_{dn}|\mu_{dn}) \tag{14}$$

Based on the introduced variational distribution, the lower bound of the log-likelihood for the review corpus $\mathcal{L}$ can be derived by the Jensen's inequality as shown below [9],

$$\log p(\mathbf{r}, \mathbf{w}, \mathbf{t}|\Phi) \geq \mathcal{L} = \mathbb{E}_q[\log p(\mathbf{r}, \mathbf{w}, \mathbf{t}, \Theta|\Phi)] - \mathbb{E}_q[\log q(\Theta|\Delta)] \tag{15}$$

المنارة للاستشارات

where the lower bound $\mathcal{L}$ consists of the expectation of log-joint distribution and the entropy of the variational distribution. Hence, we present the major component of the log-joint distribution below,

$$
\begin{aligned}
\log p(\mathbf{r}, \mathbf{w}, \mathbf{t}, \Theta|\Phi) = & \sum_{u} \log p(\pi_u^c|\alpha) + \sum_{v} \log p(\pi_v^g|\beta) + \sum_{k} \log p(\phi_k|\gamma) \\
& + \sum_{c,g} [\log p(\psi_{cg}|\eta) + \log p(\theta_{cg}|\kappa)] + \sum_{d} [\log p(c_d|\pi_{u_d}^c) \\
& + \log p(g_d|\pi_{v_d}^g) + \log p(r_d|\psi_{c_d g_d}) + \log p(t_d|\xi_{c_d g_d})] \\
& + \sum_{d,n} [\log p(z_{dn}|\theta_{c_d g_d}) + \log p(w_{dn}|\phi_{z_{dn}})]
\end{aligned}
\tag{16}
$$

Consequently, the goal is to optimize the variational parameter set Δ with the maximal lower bound $\mathcal{L}$ of the original log-likelihood, which amounts to the minimal KL divergence between the variational posterior and the true posterior. A commonly used optimization algorithm is the coordinate ascent algorithm which is an iterative fixed-point method. Particularly, we set the derivatives of Eq. 15 with respect to the variational parameters to zero accordingly and find the optimal setting for each variational parameter. In the following, we describe the details about how we update the model parameters sequentially.

Update over λ^c and λ^g For these document-dependent or word-dependent local parameters, we can just employ the updating formula from the batch variational inference. By setting the derivatives of $\mathcal{L}$ with the respect to λ^c to zero and holding other parameters fixed, we have the updating formula for λ_{di}^c indicating the probability of document d belonging to the community i

$$
\begin{aligned}
\lambda_{di}^c \propto \prod_{j} p(t_d|\xi_{ij})^{\lambda_{dj}^g} \exp \Bigg\{ & \Psi(\sigma_{u_d i}^c) + \sum_{j,s} \lambda_{dj}^g r_{ds} \left[\Psi(m_{ijs}) - \Psi\left(\sum_{s} m_{ijs} \right) \right] \\
& + \sum_{j,n,k} \lambda_{dj}^g \mu_{dnk} \left[\Psi(h_{ijk}) - \Psi\left(\sum_{k} h_{ijk} \right) \right] \Bigg\}
\end{aligned}
\tag{17}
$$

where $p(t_d|\xi_{ij})$ denotes the probability density function of the beta distribution. Similarly, we can easily derive the updating formula for λ^g below,

$$
\begin{aligned}
\lambda_{dj}^g \propto \prod_{i} p(t_d|\xi_{ij})^{\lambda_{di}^c} \exp \Bigg\{ & \Psi(\sigma_{v_d j}^g) + \sum_{j,s} \lambda_{di}^c r_{ds} \left[\Psi(m_{ijs}) - \Psi\left(\sum_{s} m_{ijs} \right) \right] \\
& + \sum_{j,n,k} \lambda_{di}^c \mu_{dnk} \left[\Psi(h_{ijk}) - \Psi\left(\sum_{k} h_{ijk} \right) \right] \Bigg\}
\end{aligned}
\tag{18}
$$

Update over μ_{dnk} For the topic distribution μ_{dnk} of the word w_{dn}, we have the updating formula below,

$$
\mu_{dnk} \propto \exp \left\{ \sum_{i,j} \lambda_{di}^c \lambda_{dj}^g \Psi(h_{ijk}) + \Psi(l_{k w_{dn}}) - \Psi\left(\sum_{x} l_{kx} \right) \right\}
\tag{19}
$$

where x is the index for the word.

المنارة للاستشارات

Update over σ^c and σ^g The optimal variational distribution for π_u^c of each user u can be estimated by the formula below [9],

$$q^{\star}(\pi_u^c|\sigma_u^c) \propto \exp\left\{\mathbb{E}_{q^{-\pi_u^c}}\left[\log p(\pi_u^c|\alpha) + \sum_{d\in D_u} \log p(c_d|\pi_{u_d}^c)\right]\right\} \tag{20}$$

where $q^{-\pi_u^c}$ denotes the variational distribution excluding π_u^c, and D_u represents the document collection for the user u. σ^c and σ^g are the global corpus-dependent parameters. Therefore, we have to take a full pass through the entire corpus for each iteration. It would be impractical to apply to large datasets. To address this problem, the SVI algorithm can easily transform the updating formula in batch setting to online minibatch setting, and the global parameters can be updated by scanning minibatch of the corpus. The updating formula for σ^c and σ^g has been shown in Eqs. 21 and 22. The analogous computation details can be found in Hoffman et al. [31].

$$\widetilde{\sigma_{ui}^c} = \alpha_i + \frac{D}{S}\sum_{d\in S_u} \lambda_{di}^c \tag{21}$$

$$\sigma_{ui}^c = (1-\rho_{\hat{t}})\sigma_{ui}^c + \rho_{\hat{t}}\widetilde{\sigma_{ui}^c} \tag{22}$$

where S is the minibatch size, and S_u represents the document collection of the user u in a minibatch. Besides, $\widetilde{\sigma_{ui}^c}$ is the optimal value if the entire corpus consists of the minibatch repeating $\frac{D}{S}$ times, and $\rho_{\hat{t}}$ is the step size for updating at the iteration $\hat{t}$. It is usually represented by an exponential decay function $(\tau_0 + \hat{t})^{\kappa_0}$. The detailed specification can be found in Hoffman et al. [31].

Due to the symmetrical form, we can easily obtain the updating formula for σ^g.

$$\widetilde{\sigma_{vi}^g} = \beta_i + \frac{D}{S}\sum_{d\in S_v} \lambda_{di}^g \tag{23}$$

$$\sigma_{vi}^g = (1-\rho_{\hat{t}})\sigma_{vi}^g + \rho_{\hat{t}}\widetilde{\sigma_{vi}^g} \tag{24}$$

Update over m, h and l Since the variational parameters m, h, and l are globally corpus-dependent, we similarly apply the SVI algorithm to derive the updating formulas for each coming minibatch, as shown below.

$$\widetilde{m_{ijs}} = \eta_s + \frac{D}{S}\sum_{d\in D_S} \lambda_{di}^c \lambda_{dj}^g r_{ds} \tag{25}$$

$$m_{ijs} = (1-\rho_{\hat{t}})m_{ijs} + \rho_{\hat{t}}\widetilde{m_{ijs}} \tag{26}$$

$$\widetilde{h_{ijk}} = \kappa_k + \frac{D}{S}\sum_{d\in D_S}\sum_{n} \lambda_{di}^c \lambda_{dj}^g \mu_{dnk} \tag{27}$$

$$h_{ijk} = (1-\rho_{\hat{t}})h_{ijk} + \rho_{\hat{t}}\widetilde{h_{ijk}} \tag{28}$$

$$\widetilde{l_{kv}} = \gamma_v + \frac{D}{S}\sum_{d\in D_S}\sum_{n} \mu_{dnk} w_{dnv} \tag{29}$$

$$l_{kv} = (1-\rho_{\hat{t}})l_{kv} + \rho_{\hat{t}}\widetilde{l_{kv}} \tag{30}$$

where D_S is the documents in a minibatch.

المنارة للاستشارات

Hyperparameter estimation According to Hoffman et al. [31], we can also incorporate updates for hyperparameters at each iteration. In terms of the hyperparameter α, the updating formula is,

$$\alpha = \alpha - \rho_{\hat{t}}\widetilde{\alpha} \tag{31}$$

where $\widetilde{\alpha}$ represents the inverse of Hessian times the gradient $\nabla_{\alpha}\mathcal{L}$. Then we can similarly present the updating formula for other hyperparameters.

$$\beta = \beta - \rho_{\hat{t}}\widetilde{\beta} \tag{32}$$

$$\eta = \eta - \rho_{\hat{t}}\widetilde{\eta} \tag{33}$$

$$\gamma = \gamma - \rho_{\hat{t}}\widetilde{\gamma} \tag{34}$$

$$\kappa = \kappa - \rho_{\hat{t}}\widetilde{\kappa} \tag{35}$$

$$\xi_{ij} = \xi_{ij} - \rho_{\hat{t}}\widetilde{\xi_{ij}} \qquad \forall i, j \tag{36}$$

We conclude the detailed algorithm in Algorithm 2.

Algorithm 2 Stochastic variational inference for TCMR model

```
Input: A collection of rating scores $r_{uv}$ given by the users $u \in \mathcal{U}$ for the items $v \in \mathcal{V}$. Each rating score $r_{uv}$
  is associated with a review text $d_{uv}$ as well as a reviewing date $t_{uv}$.
Output: Variational Parameters $\Delta = (\boldsymbol{\sigma}^{\mathbf{c}}, \boldsymbol{\sigma}^{\mathbf{g}}, \boldsymbol{\lambda}^{\mathbf{c}}, \boldsymbol{\lambda}^{\mathbf{g}}, \boldsymbol{\mu}, \mathbf{m}, \mathbf{l}, \mathbf{h})$
  Define $\rho_{\hat{t}} \triangleq (\tau_0 + \hat{t})^{-\kappa_0}$
  Initialize $\Delta$ randomly, $\alpha = \frac{1}{K_1}, \beta = \frac{1}{K_2}, \gamma = \frac{1}{V}, \kappa = \frac{1}{K}, \eta = 0.1, \xi_{ij} = (2, 2)$
  for $\hat{t} = 1 \rightarrow Max_{\hat{t}}$, each coming minibatch do
    E step:
    repeat
      Update $\lambda^c_{di}$ by Eq. 17
      Update $\lambda^g_{dj}$ by Eq. 18
      Update $\mu_{dnk}$ by Eq. 19
    until $\|$change in $\mathcal{L}(\lambda^c, \lambda^g, \mu)\| <0.00001$
    M step:
    Compute $\widetilde{\sigma^c_{ui}}, \widetilde{\sigma^g_{vi}}, \widetilde{m_{ijs}}, \widetilde{h_{ijk}}, \widetilde{l_{kv}}$ by Eq. 21, 23, 25, 27, 29
    Set $\sigma^c_{ui}, \sigma^g_{vi}, m_{ijs}, h_{ijk}, l_{kv}$ by Eq. 22, 24, 26, 28, 30
    Hyperparameter Estimation:
    Update the hyperparameters $\alpha, \beta, \eta, \gamma, \kappa, \xi$ by Eq. 31, 32, 33, 34, 35, 36
  end for
```

7 Experiment on TCMR model

In order to demonstrate the effectiveness of incorporating the time factor in TCMR, we conduct experiments on 22 larger real-world datasets with wider time span. We also compare TCMR model with the state-of-the-art time-aware approaches.

7.1 Datasets

We use the similar 22 datasets as used in the previous experiment for CMR covering different product categories from Amazon. The details can be found in Table 2. Comparing with the datasets used for CMR experiment, we remove the constraint on the number of considered

المنارة للاستشارات

Table 6 Dataset statistics for TCMR model

Dataset	No of. users	No of. items	No of. reviews	Avg. words	Time span
Amazon instant video	228, 570	21,025	463,669	142.48	Nov. 1995–Mar. 2013
Arts	24,071	4211	27,980	37.92	Apr. 1998–Mar. 2013
Automotive	133,256	47,577	188,728	38.75	Oct. 1998–Mar. 2013
Baby	123,837	6962	184,887	45.63	Feb. 1999–Mar. 2013
Beauty	167,725	29,004	252,056	37.85	Jan. 1997–Mar. 2013
Cell phones accessories	68,041	7438	78,930	50.05	Nov. 1999–Mar. 2013
Clothings	128,794	66,370	581,933	60.37	Jan. 1999–Mar. 2013
Electronics	884,175	96,643	137,1574	108.30	Nov. 1996–Mar. 2013
Gourmet foods	112,544	23,476	154,635	38.94	Jun. 1998–Mar. 2013
Health	311,636	39,539	428,781	53.28	Jul. 1998–Mar. 2013
Industrial and scientific	29,590	22,622	137,042	31.21	Aug. 1998–Mar. 2013
Jewelry	40,594	18,794	58,621	29.90	Feb. 1999–Mar. 2013
Kindle store	116,191	4372	160,793	73.48	Jul. 1995–Mar. 2013
Musical instruments	67,007	14,182	85,405	46.69	Apr. 1998–Mar. 2013
Office products	110,472	14,224	138,084	43.57	Jun. 1997–Mar. 2013
Patio	166,832	19,531	206,250	44.08	Nov. 1998–Mar. 2013
Pet supplies	160,496	17,523	217,170	43.37	Apr. 2000–Mar. 2013
Shoes	73,590	48,410	389,877	61.11	Apr. 2000–Mar. 2013
Software	68,464	11,234	95,084	63.02	Nov. 1997–Mar. 2013
Tools home improvement	290,100	53,377	419,778	88.51	Jul. 1998–Mar. 2013
Toys games	290,713	53,600	435,996	82.70	Aug. 1996–Mar. 2013
Watches	62,041	10,318	68,356	42.73	Dec. 1998–Mar. 2013

items and obtain larger datasets with wider time span. For example, we have enlarge more than ten times the size of "Clothings" dataset and widen the time span from March 2004–March 2008 to January 1999–March 2013. The dataset statistics is depicted in Table 6.

7.2 Comparative methods and evaluation metric

We compare the TCMR model with the models that achieve good performance in the previous experiments, namely, HFT and CMR. Besides, timeSVD++(TSVD) proposed by Koren [34] is the state-of-the-art time-aware collaborative filtering approach. Therefore, we compare with this model in our experiment. In terms of evaluation metric, we use the same NLL metric, as depicted in Eq. 9, to evaluate the rating prediction performance of all the approaches. Particularly, the NLL metric of TSVD can be computed by Eq. 12 with the appropriate form of predicted rating $\hat{r}_{uv}$ in TSVD.

7.3 Experimental setup

We perform similar preprocessing work as described in Sect. 5.3. In our experiment, we set the minibatch size $S = 1024$ and initialize the variational parameters Δ randomly. Each hyper-parameter is specified as: $\alpha = \frac{1}{K_1}, \beta = \frac{1}{K_2}, \gamma = \frac{1}{V}, \kappa = \frac{1}{K}, \eta = 0.1, \xi_{ij} = (2, 2), \forall i, j$. Besides, we further set $\tau_0 = 1, \kappa_0 = 0.8$ in order to guarantee convergence, which is similarly

المنارة للاستشارات

Table 7 Time-aware rating likelihood prediction performance

Method	Arts	Automotive	Baby	Beauty	Cell phones accessories	Clothings	Pet supplies	Gourmet foods	Health	Kindle store	Industrial and scientific
HFT	1.404	1.433	1.458	1.451	1.679	1.420	1.461	1.462	1.486	1.470	1.610
CMR	1.287	1.284	1.158	1.258	1.464	1.240	1.409	1.289	1.439	1.365	1.332
TSVD	1.197	1.247	1.110	1.352	1.491	1.135	1.230	1.123	1.306	1.322	1.219
TCMR	**1.119**	**1.150**	**1.097**	**1.210**	**1.413**	**1.059**	**1.156**	**1.074**	**1.219**	**1.168**	**1.176**
Method	Jewelry	Musical instrument	Office products	Patio	Shoes	Software	Amazon instant video	Tool and Home	Toys	Watches	Electronics
HFT	1.431	1.463	1.493	1.528	1.415	1.741	1.484	1.544	1.403	1.482	1.529
CMR	1.264	1.516	1.479	1.498	1.113	1.497	1.399	1.392	1.364	1.445	1.218
TSVD	1.181	1.263	1.320	1.268	1.123	1.510	1.120	1.209	1.097	1.143	1.223
TCMR	**1.130**	**1.223**	**1.295**	**1.168**	**1.015**	**1.454**	**1.083**	**1.120**	**1.071**	**1.089**	**1.160**

Lower values indicate better results

المنارة للاستشارات

Table 8 Statistical significance tests for TCMR and related models

Method	Average NLL
HFT	1.493
CMR	1.350
TSVD	1.236
TCMR	1.166†‡§

†, ‡, § Indicate that it is statistical significant at the significance level of 0.05 over HFT, CMR, and TSVD, respectively

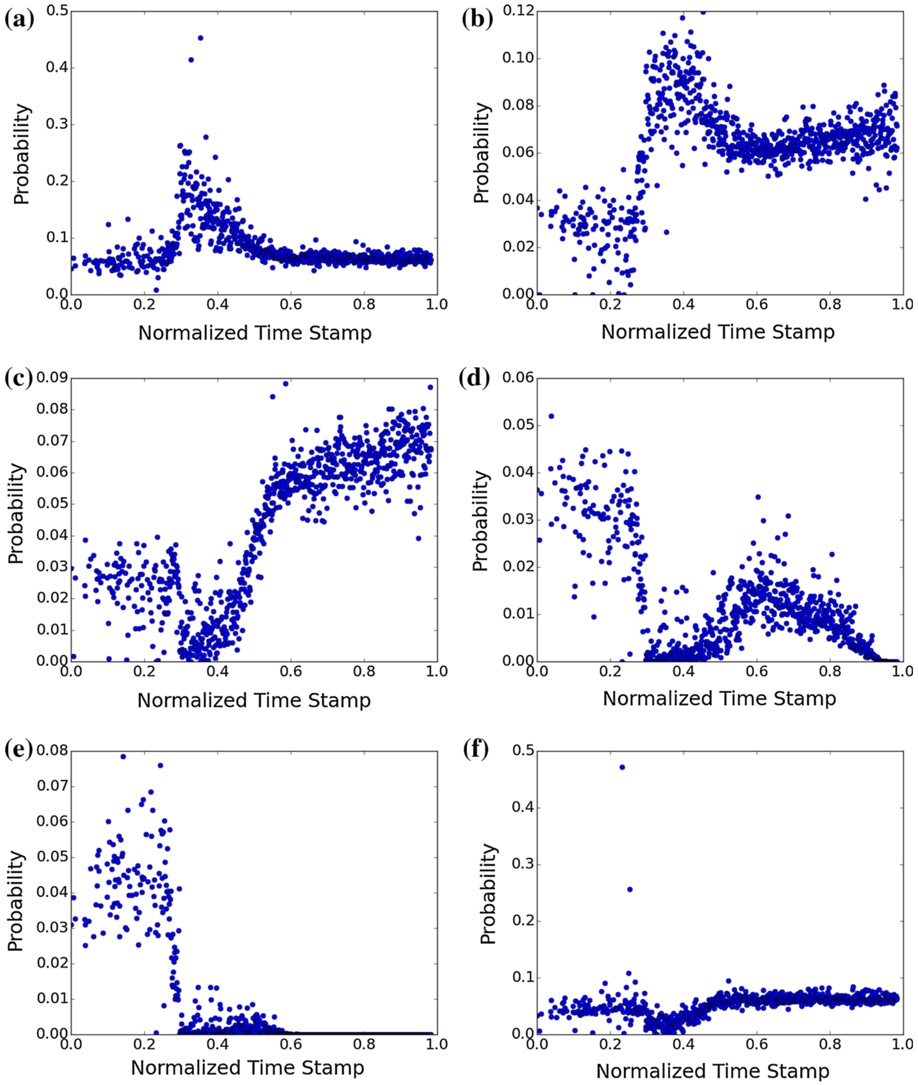


Fig. 4 Temporal change of co-clusters. **a** Prob. of a review belonging to co-cluster (1, 4) changes over time. **b** Prob. of a review belonging to co-cluster (2, 3) changes over time. **c** Prob. of a review belonging to co-cluster (4, 3) changes over time. **d** Prob. of a review belonging to co-cluster (4, 5) changes over time. **e** Prob. of a review belonging to co-cluster (5, 2) changes over time. **f** Prob. of a review belonging to co-cluster (5, 4) changes over time

المنارة للاستشارات

Table 9 Top twenty words from each topic of Shoes dataset for TCMR model

Service	Leather Shoes	Size	Appearance	Boots
Months	Leather	Size	Shoes	Boots
Pair	Pair	Fit	Comfortable	Boot
Amazon	Pretty	Shoes	Perfect	Comfortable
Buy	Toe	Wear	Style	Pair
Purchase	Sole	Bought	Love	Feet
Arrived	Foot	Wide	Look	Wear
Sneakers	Hard	Quality	Pair	Hiking
Time	Pairs	Return	Wear	Support
Received	Brown	Half	Price	Break
Sandals	Black	Feet	Shoe	Price
Shipping	Arch	Look	Walking	Warm
Store	Stars	Narrow	Looking	Waterproof
Money	Sore	Tight	Recommend	Socks
Keds	Rubber	Slippers	Colors	Walking
Service	Shoes	Foot	Happy	Recommend
Online	Top	Heel	Feet	Worn
Brand	Soft	Leather	Running	Light
Days	Satisfied	Width	Bought	Fit
Purchased	Wider	True	White	Time
Worth	Worn	Sole	Worn	Wet

done in Hoffman et al. [31]. We construct the training set by the beginning 80% according to the time of the reviews. We select the last 10% of the reviews to form test set. The remaining middle 10% of the reviews is used for parameter tuning (validation set).

7.4 Results on rating prediction

In our proposed TCMR model, we also fix the number of topics K as 5 and conduct the grid search for the number of user community K_1 and item group K_2 in the range of [1, 15] and [1, 10], respectively, using the validation set. For each dataset, we determine K_1, K_2 with the lowest NLL on the validation set and then calculate the corresponding NLL metric on the test set. For the comparative methods, the parameter setting we use is the same as described in their papers [34,40,64], and we also calculate the NLL metric on the same test set.

The results are reported in Table 7, and the best performance has been bolded. As depicted in the results, the TCMR outperforms all the comparative methods on all 22 datasets. Comparing with our previous CMR model and the HFT model, TCMR generally performs better because user's rating behavior and concerned topics are drifting over time. Therefore, time consideration is desirable for the dataset with wider time span. Besides, another observation is that TCMR has superior rating prediction performance than the state-of-the-art time-aware TSVD method demonstrating the usefulness of considering the review texts as well as the hidden user community and item groups. We conduct the t test for the results with the significance level being 0.05. The test results are reported in Table 8. It is statistical significant that TCMR performs better than all the other methods.

المنارة للاستشارات

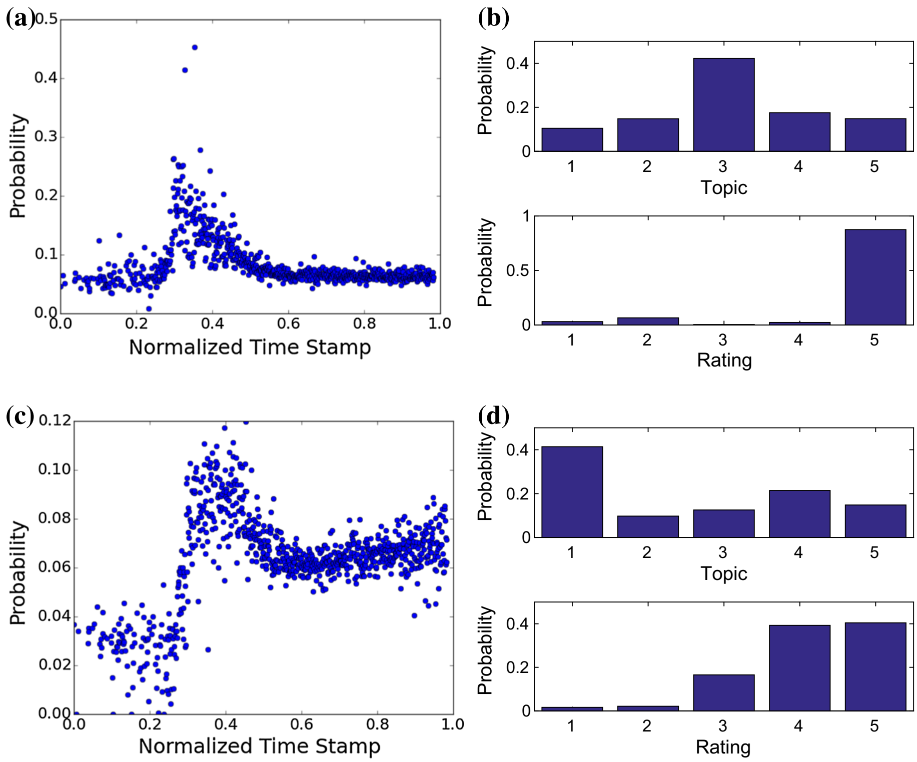


Fig. 5 Topic distribution and rating distribution of two typical community–group co-clusters on Shoes dataset. **a** Prob. of a review belonging to co-cluster (1, 4) changes over time. **b** Topic distribution and rating distribution of co-cluster (1, 4). **c** Prob. of a review belonging to co-cluster (2, 3) changes over time. **d** Topic distribution and rating distribution of co-cluster (2, 3)

7.5 Temporal effect on the user–item co-clusters

We investigate the temporal effect on the co-clusters of user communities and item groups. The dataset we use is "Shoes" category from Amazon since our proposed TCMR has achieved the best rating prediction performance. For our TCMR model, we fix the number of topics K as 5 and equally specify the number of user community K_1 and item group K_2 with 5. Then we perform the rating prediction experiment and obtain the user community mixture membership vector λ_d^c and item group mixture membership vector λ_d^g for each review d. As a result, we can figure out the average $\overline{\lambda_d^c}$ and $\overline{\lambda_d^g}$, $\forall d \in D_t$, for all the reviews D_t at each reviewing time t, and obtain a probability matrix $P_t =< \overline{\lambda_d^c}, \overline{\lambda_d^g}^T >$ with the entry $P_t(i, j)$ indicating the average probability of a certain review belonging to the co-cluster (i, j). The plots in Fig. 4 capture the dynamic change of several typical co-clusters over the time. For each co-cluster (i, j) in Fig. 4, the x-axis represents the time and the y-axis denotes the average probability of a certain review belonging to the co-cluster (i, j) which can be interpreted as the popularity of a co-cluster at a given time.

Figure 4a shows that a certain small user community tends to buy a sort of item group at the beginning, and then more and more people become the fans of such item group possibly during the promotion period. However, consumers' enthusiasm may disappear after the promotion,

المنارة للاستشارات

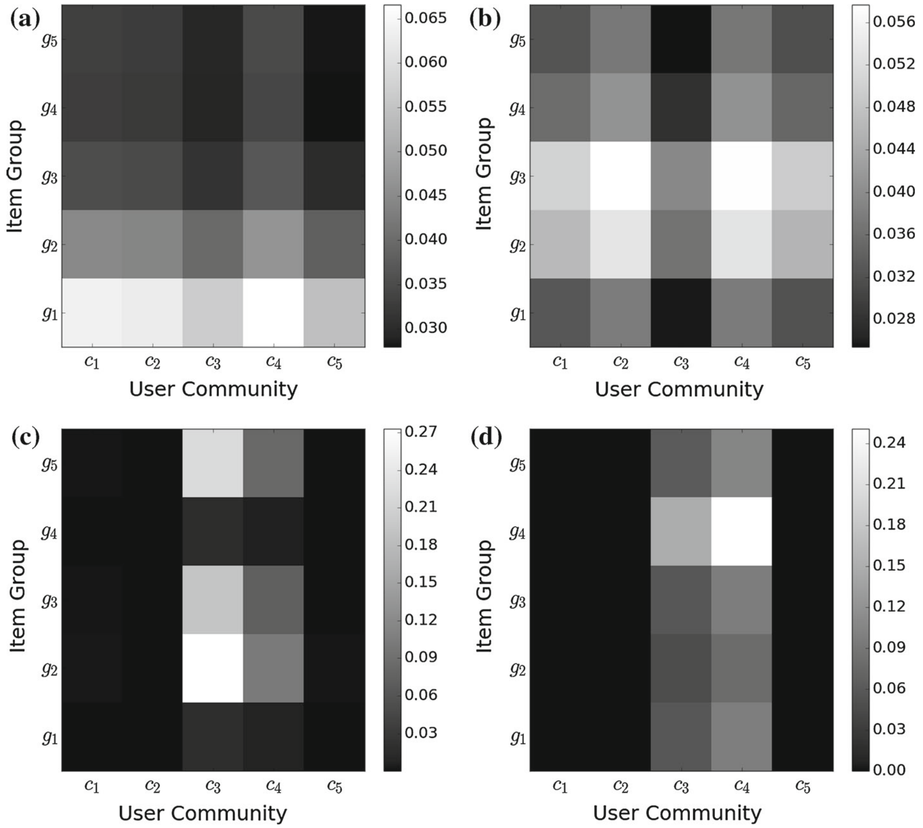


Fig. 6 The heatmap of the co-cluster at different time stamp. **a** The Avg. Prob. of a review belonging to each co-cluster at $t = 0.050$. **b** The Avg. Prob. of a review belonging to each co-cluster at $t = 0.193$. **c** The Avg. Prob. of a review belonging to each co-cluster at $t = 0.382$. **d** The Avg. Prob. of a review belonging to each co-cluster at $t = 0.670$

and the scale of such user community returns to its original level. Different from Fig. 4a, Fig. 4b shows us the continuous positive effect of product item promotion. A certain amount of users stay within the fans community of such item group even after the promotion period. Besides, Fig. 4c presents another trend for the co-cluster popularity change. Some users originally have good impressions for this kind of item, but its reputation on the product quality dropped so that users hardly bought it. After a certain period, the sales of such item have been recovered and even become better. The fourth trend in Fig. 4d for the co-cluster (4, 5) is a bit different from Fig. 4c. In this situation, the advertising investment did not help recover the sales of the product item, but most of the consumers left after fully digesting the advertisement. We also show a case in Fig. 4e that any strategies are not useful for vitalizing the sales of the product item after a dramatic fall. Finally, the last Fig. 4f reports a relatively stable users' inclination for the product item.

Furthermore, we also present the five discovered topics by our model TCMR as shown in Table 9. The interpretation for each topic is similar with the topics discovered by the CMR model in Table 5b. Then we select co-cluster (1, 4) and (2, 3) in Fig. 4a, b as an example to show their corresponding topic distribution and rating distribution in Fig. 5. Specifically, users from the co-cluster (1, 4) tend to discuss the third topic, namely, "size" of the shoes in

المنارة للاستشارات

their reviews, and they always provide the highest five star ratings according to Fig. 5b. As shown in Fig. 5a, such phenomenon has a burst in the middle of the time span and then returns to its original level after a certain period. Similarly, as shown in Fig. 5d, co-cluster (2, 3) is characterized by the topic distribution with the focus on the first topic "Service" and the rating distribution with the major components of four and five stars. Figure 5c demonstrates that such discussed topics and rating behavior also have a burst in the middle of the time span and then still keep a high level in the later stage.

Moreover, in Fig. 6, we also provide the heat map for co-cluster probability matrix P at some typical time stamps. For each sub-figure in Fig. 6, the x-axis denotes five user communities (c_1, c_2, c_3, c_4, and c_5) and the y-axis represents five item groups (g_1, g_2, g_3, g_4, and g_5). Each cell (c_i, g_j) indicates the average probability of a review belonging to co-cluster (c_i, g_j). According to the greyscale bar on the right, cells with the color close to white have a larger probability value while cells with the color close to black have a smaller probability value. As depicted in Fig. 6a, at the time of $t = 0.050$, the main co-clusters are (c_1, g_1), (c_2, g_1) and (c_4, g_1), which indicates that the item group of g_1 is more popular than others. Nevertheless, the most popular item groups turns to g_2 and g_3 when the time comes to $t = 0.193$ as shown in Fig. 6b. After that, in Fig. 6c the co-cluster (c_3, g_2) dominates at the time of $t = 0.382$, and Fig. 6d of $t = 0.670$ has the major co-cluster (c_4, g_4). From the discussion above, we can clearly observe that the co-clusters of hidden user community and item group change over the time, and the rating prediction result can be improved with the help of exploiting such temporal effect.

8 Conclusions

We have proposed a new generative model CMR to predict user's ratings on previously unrated items by incorporating the review texts and hidden user community and item group information into a collaborative filtering method. Due to the dyadic characteristic of the input user–item rating matrix, co-clustering technique is employed to model the relationship of hidden user communities and item groups. We have performed extensive experiments on 22 real-world datasets. The experimental results show that our model CMR outperforms the state-of-the-art methods. Furthermore, we extend our CMR model to TCMR model (Time-aware CMR) by considering time information and exploiting the temporal interaction among review texts and co-clusters of user communities and item groups. In this TCMR model, each community–group co-cluster is characterized by an additional beta distribution for time modeling. To evaluate our TCMR model, we have conducted another set of experiments on 22 larger datasets with wider time span. Our proposed TCMR model performs better than CMR and the standard time-aware recommendation model on the rating score prediction tasks. We also investigate the temporal effect on the user–item co-clusters.

Acknowledgements The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14203414) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055034). This work is also affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies.

References

1. Agarwal D, Chen B-C (2010) flda: matrix factorization through latent dirichlet allocation. In: Proceedings of the 3rd ACM international conference on web search and data mining, pp 91–100

المنارة للاستشارات

2. Agarwal D, Merugu S (2007) Predictive discrete latent factor models for large scale dyadic data. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, pp 26–35
3. Aizenberg N, Koren Y, Somekh O (2012) Build your own music recommender by modeling internet radio streams. In: Proceedings of the 21st international conference on world wide web, pp 1–10
4. Almahairi A, Kastner K, Cho K, Courville A (2015) Learning distributed representations from reviews for collaborative filtering. In: Proceedings of the 9th ACM conference on recommender systems, pp 147–154
5. Baltrunas L, Amatriain X (2009) Towards time-dependant recommendation based on implicit feedback. In: Workshop on context-aware recommender systems
6. Bao Y, Fang H, Zhang J (2014) Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In: Proceedings of the 28th AAAI conference on artificial intelligence, pp 2–8
7. Bell R, Koren Y, Volinsky, C (2007) Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: Proceedings of the 13rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 95–104
8. Beutel A, Murray K, Faloutsos C, Smola A (2014) Cobafi: collaborative bayesian filtering. In: Proceedings of the 23rd international conference on world wide web, pp 97–108
9. Bishop CM (2006) Pattern recognition and machine learning. Information science and statistics. Springer, New York
10. Blei D, Ng A, Jordan M (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
11. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132
12. Cai Y, Leung H-F, Li Q, Min H, Tang J, Li J (2014) Typicality-based collaborative filtering recommendation. IEEE Trans Knowl Data Eng 26(3):766–779
13. Chen C, Li D, Zhao Y, Lv Q, Shang L (2015) Wemarec: accurate and scalable recommendation through weighted and ensemble matrix approximation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp 303–312
14. Chen T, Han W, Wang H, Zhou Y, Xu B, Zang B (2007) Content recommendation system based on private dynamic user profile. In: IEEE international conference on machine learning and cybernetics, pp 2112–2118
15. Cheng Y, Church G (2000) Biclustering of expression data. In: Proceedings of the 8th international conference on intelligent systems for molecular biology, vol 8, pp 93–103
16. Chu W, Park S (2009) Personalized recommendation on dynamic content using predictive bilinear models. In: Proceedings of the 18th international conference on world wide web, pp 691–700
17. DeCoste D (2006) Collaborative prediction using ensembles of maximum margin matrix factorizations. In: Proceedings of the 23rd international conference on machine learning, pp 249–256
18. Dhillon I, Mallela S, Modha D (2003) Information-theoretic co-clustering. In: Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining, pp 89–98
19. Diao Q, Qiu M, Wu C-Y, Smola AJ, Jiang J, Wang C (2014) Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 193–202
20. Ding Y, Li X (2005) Time weight collaborative filtering. In: Proceedings of the 14th ACM international conference on information and knowledge management, pp 485–492
21. Gaillard J, Renders J-M (2015) Time-sensitive collaborative filtering through adaptive matrix completion. In: Proceedings of the 37th European conference on information retrieval, pp 327–332
22. Ganu G, Elhadad Y, Marian A (2009) Beyond the stars: improving rating predictions using review text content. In: Proceedings of the 12th international workshop on the web and databases
23. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6:721–741
24. George T, Merugu S (2005) A scalable collaborative filtering framework based on co-clustering. In: Proceedings of the 5th IEEE international conference on data mining, pp 625–628
25. Geuens S (2015) Factorization machines for hybrid recommendation systems based on behavioral, product, and customer data. In: Proceedings of the 9th ACM conference on recommender systems, pp 379–382
26. Griffiths T, Steyvers M (2004) Finding scientific topics. In: Proceedings of the National academy of Sciences of the United States of America, pp 5228–5235
27. Guan L, Alam MH, Ryu W, Lee S (2016) A phrase-based model to discover hidden factors and hidden topics in recommender systems. In: IEEE international conference on big data and smart computing (BigComp), pp 337–340
28. Hartigan J (1972) Direct clustering of a data matrix. J Am Stat Assoc 67(337):123–129

المنارة للاستشارات

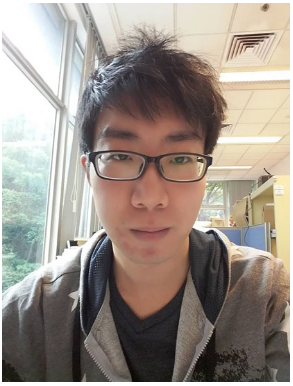
29. He X, Chen T, Kan M, Chen X (2015) Trirank: review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM international conference on information and knowledge management, pp 1661–1670
30. Heckel R, Vlachos M (2016) Interpretable recommendations via overlapping co-clusters. arXiv preprint arXiv:1604.02071
31. Hoffman M, Bach F, Blei D (2010) Online learning for latent dirichlet allocation. In: Proceedings of advances neural information processing systems, pp 856–864
32. Hong L, Doumith A, Davison B (2013) Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In: Proceedings of the sixth ACM international conference on web search and data mining, pp 557–566
33. Koren Y (2008) 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
34. Koren Y (2010) Collaborative filtering with temporal dynamics. Commun ACM 53(4):89–97
35. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
36. Lacoste-Julien S, Sha F, Jordan M (2008) Disclda: discriminative learning for dimensionality reduction and classification. In: Proceedings of advances neural information processing systems, vol 83, p 85
37. Lathia N, Hailes S, Capra L (2009) Temporal collaborative filtering with adaptive neighbourhoods. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp 796–797
38. Loni B, Shi Y, Larson M, Hanjalic A (2014) Cross-domain collaborative filtering with factorization machines. In: 36th European conference on information retrieval, pp 656–661
39. McAuley J, Leskovec J (2013) From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd international conference on world wide web, pp 897–908
40. McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on recommender systems, pp 165–172
41. McAuley J, Leskovec J, Jurafsky, D (2012) Learning attitudes and attributes from multi-aspect reviews. In: Proceedings of the 12th IEEE international conference on data mining, pp 1020–1025
42. Paterek A (2007) Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD cup and workshop, PP 5–8
43. Qiang R, Liang F, Yang J (2013) Exploiting ranking factorization machines for microblog retrieval. In: Proceedings of the 22nd ACM international conference on information and knowledge management, pp 1783–1788
44. Rendle S (2012) Factorization machines with libfm. ACM Trans Intell Syst Technol (TIST) 3(3):57
45. Rennie J, Srebro N (2005) Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd international conference on machine learning, pp 713–719
46. Salakhutdinov R, Andriy M (2007) Probabilistic matrix factorization. In: Proceedings of advances neural information processing systems, vol 1, pp 1–2
47. Salakhutdinov R, Andriy M (2008) Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proceedings of the 33rd international conference on machine learning, pp 880–887
48. Sarwar B, Karypis G, Konstan J, Riedl J (2002) Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Proceedings of 5th international conference on computer and information science, pp 27–28
49. Shan H, Banerjee A (2008) Bayesian co-clustering. In: Proceedings of the 8th IEEE international conference on data mining, pp 530–539
50. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput Surv (CSUR) 47(1):3
51. Srebro N, Alon N, Jaakkola T (2004) Generalization error bounds for collaborative prediction with low-rank matrices. In: Proceedings of advances neural information processing systems
52. Srebro N, Jaakkola T (2003) Weighted low-rank approximations. In: Proceedings of the 20th international conference on machine learning, vol 3, pp 720–727
53. Srebro N, Rennie J, Jaakkola T (2004) Maximum-margin matrix factorization. In: Proceedings of advances neural information processing systems, vol 17, pp 1329–1336
54. Tan C, Chi E, Huffaker D, Kossinets G, Alexander S (2013) Instant foodie: predicting expert ratings from grassroots. In: Proceedings of the 22nd ACM international conference on information and knowledge management, pp 1127–1136
55. Titov I, McDonald R (2008) A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of the 46th annual meeting of the Association for Computational Linguistics, pp 308–316

المنارة للاستشارات

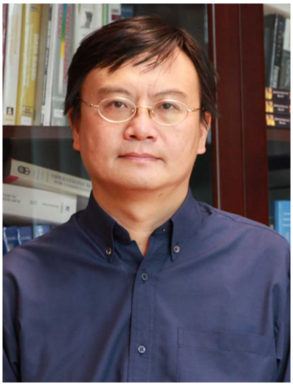
56. Titov I, McDonald R (2008) Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th international conference on world wide web, pp 111–120
57. Wang C, Blei D (2011) Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 448–456
58. Wang X, McCallum A (2006) Topics over time: a non-markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 424–433
59. Weimer M, Karatzoglou A, Le Q, Smola A (2007) Maximum margin matrix factorization for collaborative ranking. In: Proceedings of advances neural information processing systems, pp 1593–1600
60. Weimer M, Karatzoglou A, Smola A (2008) Improving maximum margin matrix factorization. Mach Learn 72(3):263–276
61. Weston J, Bengio S, Usunier N (2011) Wsabie: Scaling up to large vocabulary image annotation. In: Proceedings of the 22nd international joint conference on artificial intelligence, pp 2764–2770
62. Xiang L, Yuan Q, Zhao S, Chen L, Zhang X, Yang Q, Sun J (2010) Temporal recommendation on graphs via long-and short-term preference fusion. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, pp 723–732
63. Xin X, Liu Z, Lin C, Huang H, Wei X, Guo P (2015) Cross-domain collaborative filtering with review text. In: Proceedings of the 24th international joint conference on artificial intelligence, pp 1827–1833
64. Xu Y, Lam W, Lin T (2014) Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups. In: Proceedings of the 23rd ACM international conference on information and knowledge management, pp 251–260
65. Yang S, Long B, Alexander S, Zha H, Zheng Z (2011) Collaborative competitive filtering: learning recommender using context of user choice. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, pp 295–304
66. Yin H, Cui B, Chen L, Hu Z, Zhou X (2015) Dynamic user modeling in social media systems. ACM Trans Inf Syst (TOIS) 33(3):10
67. Yu J, Shen Y, Yang Z (2014) Topic-stg: Extending the session-based temporal graph approach for personalized tweet recommendation. In: Proceedings of the companion publication of the 23rd international conference on world wide web companion, pp 413–414
68. Yu K, Zhu S, Lafferty J, Gong Y (2009) Fast nonparametric matrix factorization for large-scale collaborative filtering. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp 211–218
69. Zhang T, Iyengar VS (2002) Recommender systems using linear classifiers. J Mach Learn Res 2:313–334
70. Zhang Y, Lai G, Zhang M, Zhang Y, Liu Y, Ma S (2014) Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 83–92
71. Zhang Y, Zhang M, Zhang Y, Lai G, Liu Y, Zhang H, Ma S (2015) Daily-aware personalized recommendation based on feature-level time series analysis. In: Proceedings of the 24th international conference on world wide web, pp 1373–1383
72. Zimdars A, Chickering DM, Meek C (2001) Using temporal data for making recommendations. In: Proceedings of the 7th conference on uncertainty in artificial intelligence, pp 580–588

Yinqing Xu received a B.E. degree in the Department of Computer Science and Technology from Northwestern Polytechnical University, Xi'an, China, in 2012. He is currently a Ph.D. student at the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong SAR, China. His research interests include text mining, recommendation system and machine learning.

المنارة للاستشارات



Qian Yu received B.E. degree and M.S. degree in the School of Computer Software from Tianjin University, Tianjin, China, in 2012 and 2015. He is currently a Ph.D. student at the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong SAR, China. His research interests include text mining and machine learning.



Wai Lam received a Ph.D. in Computer Science from the University of Waterloo. He obtained his B.Sc. and M.Phil. degrees from The Chinese University of Hong Kong. After completing his Ph.D. degree, he conducted research at Indiana University Purdue University Indianapolis (IUPUI) and the University of Iowa. He joined The Chinese University of Hong Kong, where he is currently a professor. His research interests include intelligent information retrieval, text mining, digital library, machine learning, and knowledge-based systems. He has published articles in IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Information Systems, etc. His research projects have been funded by the Hong Kong SAR Government General Research Fund (GRF) and DARPA (USA). He also managed industrial projects funded by Innovation and Technology Fund (industrial grant) and IT companies.

Tianyi Lin received a B.S. degree from Nanjing University, Nanjing, China, in 2011, and an M.S. degree from University of Cambridge, Cambridgeshire, UK, in 2012. He has been a junior research assistant in the Chinese University of Hong Kong since 2013. His current research interests include numerical optimization, statistics, machine learning, and text mining.

المنارة للاستشارات

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.

المنارة للاستشارات