Ratings Meet Reviews, a Combined Approach to Recommend

Guang Ling[1,2], Michael R. Lyu[1,2] and Irwin King[1,2]
[1]Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications
Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China
[2]Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
gling@cse.cuhk.edu.hk, lyu@cse.cuhk.edu.hk, king@cse.cuhk.edu.hk

ABSTRACT

Most existing recommender systems focus on modeling the ratings while ignoring the abundant information embedded in the review text. In this paper, we propose a unified model that combines content-based filtering with collaborative filtering, harnessing the information of both ratings and reviews. We apply topic modeling techniques on the review text and align the topics with rating dimensions to improve prediction accuracy. With the information embedded in the review text, we can alleviate the cold-start problem. Furthermore, our model is able to learn latent topics that are interpretable. With these interpretable topics, we can explore the prior knowledge on items or users and recommend completely "cold" items. Empirical study on 27 classes of real-life datasets show that our proposed model lead to significant improvement compared with strong baseline methods, especially for datasets which are extremely sparse where rating-only methods cannot make accurate predictions.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information filtering; I.2.6 [**Artificial Intelligence**]: Learning-parameter learning

Keywords

Collaborative Filtering; Content-based Filtering; Cold-start Problem

1. INTRODUCTION

With the ever growing number of choices available online, recommender systems are becoming more and more indispensable. We rely on them to select favorite songs from millions of collections in music streaming services like Spotify and iTunes Radio. We depend upon them to suggest interesting movies in movie rating website such as IMDb and video streaming providers such as Netflix. Amazon uses recommender systems to suggest products to potential users. Now the company even takes it to another level by shipping the products to the warehouse near the customer based on speculated order produced by recommender systems[1].

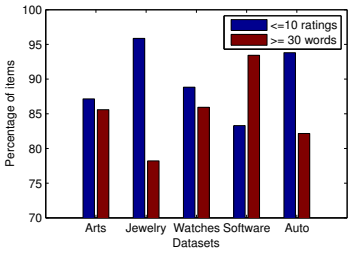


Figure 1: Percentage of items having less than 10 ratings and more than 30 words in various Amazon datasets

Although recommender systems employed in industry seem to perform well in practice, there are some deficiencies with existing approaches. The first problem confronted with most recommender system is their inability to deal with so called *cold-start* problem [28]. When a new user joins a recommender system, there is little data available for the system to learn the preferences of the user accurately. Without an accurate representation of the user, the system cannot make recommendations confidently. Similarly, the systems defer the recommendations for newly included items as well. The cold-start problem leads to poor experience for new users and also when recommending new items. In real-life recommender systems, the cold-start problem is a severe problem. Shown in Figure 1 are the statistics of 5 categories in Amazon datasets [16]. Across the 5 listed datasets, over 80% of items have few ratings (less than 10). While at the same time, over 70% of items have review text at length (over 30 words). The ratings alone are inadequate to learn the preferences accurately. Review comments complement the ratings by providing rich knowledge of the items and preferences of the users. Harnessing the information embedded in the review text is the key to successful recommendation in such scenarios.

Another drawback of existing recommender systems is their poor interpretability, making further understanding of users' preference as well as items' properties impossible. For example in matrix-factorization [25] based methods, we learn two latent feature matrices corresponding to users' latent features and items' latent features. The dot product between a user's and an item's feature vector is used to predict the rating that the user would assign to the item. It is challenging to associate these real valued features with con-

[1]http://on.rt.com/c8v82n

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

http://dx.doi.org/10.1145/2645710.2645728.

ceivable physical meanings. We know that a user might like an item due to a particular latent feature since they both have a large positive (or negative) value on that feature. But we have no clue of the feature's physical meaning. Does it mean that the user is fond of Sci-Fi and the movie belongs to Sci-Fi genre? Or is it that the user loves the leading actor of the movie? We do not know. In fact, it is possible that each feature corresponds to a combination of the human interpretable features, rendering the feature interpretation problem more difficult.

Both of the above problems can be solved or at least alleviated by combining *content-based filtering* and *collaborative filtering*. In collaborative filtering, we make predictions on a user's preferences over items based on all users' past ratings. Collaborative filtering rooted in the keen observation that users who shared similar preferences in the past tend to rate similarly in the future. A collaborative filtering model uses only the past rating information and does not take the contents of the items into consideration. On the other hand, content-based filtering approaches the recommendation problem by analyzing the content of the items and matches it with the preference of a user.

In a recommender system, apart from an integer score, users are often allowed to write text reviews about the item to complement the rating. The review text contains a source of rich information *explaining* the reason why the user assigns such a rating to the item. These reviews provide text contents of the items, which can be leveraged to alleviate cold-start problem when the ratings are sparse. This is because the information embedded in the text review is much richer that an integer rating. When we have few ratings, it is nearly impossible to learn an accurate feature of the concerned user/item. However, the text review might allow us to better estimate the features. To solve the interpretation problem, we align latent topic spaces with the rating spaces. Each latent dimension is tagged with a word cloud, explaining the physical meaning of the dimension. For example, when we see that a user and a movie have large positive value on the third feature, which has text label "thriller, sci-fi, nolan", we know that this user likes the science fiction thriller movie directed by Christopher Nolan.

Interpretability and the cold-start problem are not two isolated problems. Learning an interpretable model could help alleviate the cold-start problem [1, 13]. We can leverage prior knowledge of items and suggest completely "cold" items with confidence. For example, if we know that a user assigns high scores for the topic tagged with "fantasy, adventure, peter, jackson", a recommender system can confidently recommend "The Hobbit: There and Back Again" (a fantasy adventure movie directed by Peter Jackson) to the user even if this movie is not being shown yet.

The contribution of this paper is three-fold. First, we propose a novel method to combine content-based filtering seamlessly with collaborative filtering, modeling the reviews and ratings simultaneously. Secondly, we derive an efficient collapsed Gibbs sampling method to learn the model. Thirdly, we demonstrate our model's advantage in prediction accuracy compared with previous work, especially under the cold-start setting, on large real-life datasets with millions of users and items. We also show the interpretability of the model using a few examples.

2. RELATED WORK

Recommender systems can broadly be classified into two types: *content-based filtering* [21] and *collaborative filtering*. Content-based filtering recommends an item to a user by matching up the features of the item with the preferences of the user, both of which are learnt by analyzing the contents and profiles. Collaborative filtering (CF), on the other hand, approaches the recommendation problem by analyzing the co-occurrence patterns of user-item pair, which is often attached with an integer rating. There are extensive investigations on collaborative filtering, from neighborhood-based methods [27, 11] to model-based methods [9, 26]. Recently, some methods concentrated on ranking [10, 24] the items better. Other approaches leveraged social [14, 15] and side information [32] to improve the performance.

Due to the advantage of taking the review text into consideration, there are a few efforts [2, 4, 16, 17] explored the combination of content-based filtering and collaborative filtering. In the early work [17], the authors cast the content-based filtering as a classification problem, using which they filled out some of the unobserved user item rating matrix and apply collaborative filtering methods on this denser matrix. The authors of [2] cast the recommendation problem as an ordinal regression problem and applied a combination of kernels to handle the side information. In [4], the authors found that there were a few aspects that affect how users rate items. They harnessed the information embedded in the review text to learn how a user weights these aspects and how an item distributes on these aspects. However, their method required human annotators with expert domain knowledge to pre-define these aspects rather than learning them automatically from the reviews.

In the recent work [16], the authors proposed the Hidden Factors and Hidden Topics (HFT) model, which learnt a Latent Dirichlet Allocation (LDA) [3] model for items using the review text and a matrix factorization model to fit the ratings. To bridge the gap between the stochastic vector obtained from LDA and the real-valued vector in MF model, the authors proposed a transformation to link the two. Their method demonstrated significant improvement over baseline methods that use ratings or reviews alone. However, the transformation function they employ, the exponential function, fixed the relationship between latent vector in MF and the topic distribution. Although a parameter is employed to maintain a more flexible relationship, it is still difficult to ensure that this transformation is correctly scaled.

In [8, 29, 30], the authors also considered the interpretable aspects to make better predictions. However, their approaches differ from ours in that either they had *explicit* ratings per aspect (i.e., multiple ratings on prescribed aspects per user item pair), or these aspects were inferred from other context than review text. In another line of research called sentiment analysis [6, 12], a positive or negative label rather than an integer score is learnt for a short text. Our work also differs from [18], which recommends personalized reviews. In [31], the authors proposed Collaborative Topic Regression (CTR) to suggest scientific articles to potential readers. Later work [22] extended it to take the social network among users into consideration. As pointed out in [16], the latent dimensions they discovered were not necessarily correlated with ratings.

3. RATINGS MEET REVIEWS

Our model, titled "Ratings Meet Reviews" (RMR), is a probabilistic generative model that combines a topic model seamlessly with a rating model. We describe it as follows.

3.1 Model and Notations

Suppose there are N users $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$, M items $\mathcal{V} = \{v_1, v_2, \cdots, v_M\}$, a set of observed indices $Q = \{i, j\}$, where $\{u_i, v_j\} \in \mathcal{U} \times \mathcal{V}$ defines the observed ratings $\mathcal{X} = \{x_{i,j}\}$, each of which is optionally associated with a review $r_{i,j} = \{w | w \subset V\}$ of length $L_{i,j}$, where V is the set of vocabulary used in the review text. Alternatively, let $\mathcal{U}_j$ denote the indices of users who have rated item v_j. Let K denote the number of topics.

RMR operates on items[2], which has an intrinsic distribution θ on topics. This distribution describes the proportion that the item belongs to each topic.

We present the generative process below:

1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.

From the generative process, we can identify that the text reviews are generated similarly as the LDA model. We use a mixture of Gaussian rather than matrix factorization based methods [16, 31] to model the ratings. These user-topic specific Gaussian distributions have clear interpretations. They describe how a user values the aspects denoted by each latent topic. The item is modeled as a distribution of topics, which together with the user-topic specific Gaussian distributions determine how a user would rate the item. The ratings and review text are connected by the same item topic distribution θ. The more a user talks about certain aspects concerning an item, the higher the distribution will be on these topics, which in turn affects the rating that the user would assign to the item.

We choose to model ratings using mixture of Gaussians for two reasons. First, we can avoid the difficult choice of the transformation function employed in [16]. As discussed above, the transformation function is restrictive and the scaling parameter is non-trivial to select. Secondly, we can retain the interpretability of the topics with no compromise. The interpretability of the latent dimensions is an important factor to solve the cold start problem. Take book recommendation for example, when a user showed strong interest in dimension with high probability on words "da vinci code dan brown". We can confidently recommend Dan Brown's new book "Inferno". We are able to associate the latent dimensions with the *prior* knowledge (for example, Meta data) that is available *without* ratings or reviews.

Given the generative process, the probability of observing the review text and the ratings given the model parameters $\Theta = \{\theta, \psi, \mu\}$ is

$$P(\mathbf{w}, \mathbf{x}|\Theta; \alpha, \beta, \mu_0, \sigma_0^2, \sigma^2) = \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i \in \mathcal{U}_j} \left(\sum_{f=1}^{K} P(f|\theta_j) P(x_{i,j}|\mu_{i,f}, \sigma^2) \right) \left(\prod_{l=1}^{L_{i,j}} \sum_{z=1}^{K} P(z|\theta_j) P(w_l|\psi_z) \right) \left(\prod_{i=1}^{N} \prod_{k=1}^{K} P(\mu_{i,k}|\mu_0, \sigma_0^2) \right) \left(\prod_{k=1}^{K} P(\psi_k|\beta) \right). \quad (1)$$

If we take the log of Eq. (1), we get the log-likelihood of model parameters. However, because of the summation inside the log, direct optimization is not feasible. We subsequently develop an efficient collapsed Gibbs sampling method to learn the model parameters in Section 3.3. We now take a deeper look at RMR and compare it with HFT and CTR.

3.2 Comparison with HFT and CTR

Shown in Figure 2 are the graphical models of RMR and several related work. As is clear from the figure, the left parts of CTR, HFT and RMR resemble LDA, which was originally proposed by David Blei et al. to learn the latent topics in a corpus of documents (items in our setting) in an unsupervised manner. The LDA algorithm assumes there are K latent topics in the corpus, which are K multinomial distributions over the vocabulary. Each document in the corpus is a mixture of these topics. A document specific topic distribution over the K latent topics, θ_j with Dirichlet prior α, governs how much weight each topic takes in document j. This θ_j is a length-K stochastic vector with non-negative entries which sums up to 1.

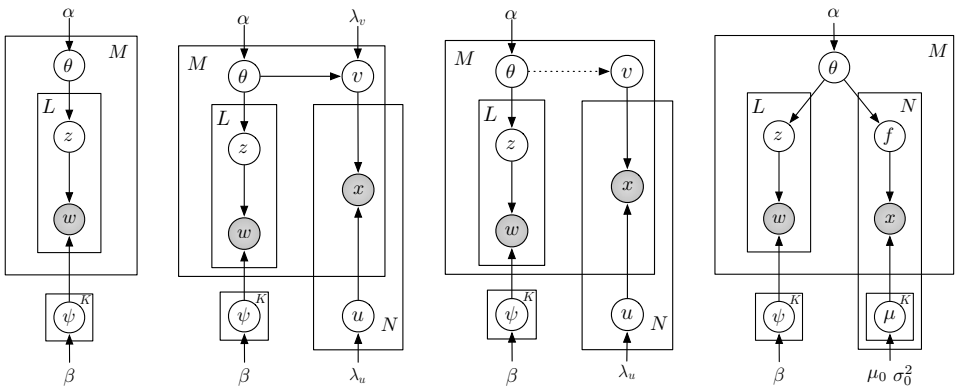
Both HFT and CTR adopt the matrix factorization method to model the ratings. Arrange users' ratings on items in a partially observed matrix $X \in \mathbb{R}^{N \times M}$. The matrix factorization model assumes that X has a low rank structure and thus can be decomposed in to the product of two matrices $U^T V$, where both U and V are of rank $K \ll \min(M, N)$. The columns of U and V can be interpreted as the latent features of users and items respectively. The dot product between a user's feature vector and an item's feature vector approximates the rating that the user would assign to the item. To regularize the value that the latent features can assume, often zero-mean isotropic Gaussian priors are placed on both the user and item latent features. The objective function of a matrix factorization model can be formulated as follows,

$$\mathcal{L} = \sum_{i,j \in Q} (U_i^T V_j - X_{i,j})^2 + \lambda_U \|U\|_F + \lambda_V \|V\|_F, \quad (2)$$

in which the first term is the difference between observed and prediction and the rest are regularization terms.

Clearly, there is a discrepancy between the item topic distribution θ_j in LDA and the item feature vector V_j in MF model. The former is a distribution which is all positive and sums up to 1 while the later can assume any real value. Both HFT and CTR try to align the item features with the item topic distribution and hence the rich text review can be exploited to better model the item features. The main difference between HFT and CTR model lies in the way they align the topic distribution θ_j and the item feature V_j. In CTR, the item feature V_j is assumed to be a Gaussian random variable with mean θ_j and precision c. In other words, the θ_j is taken as the *default* value of V_j, but the later can adapt to match the ratings. If an item receives a lot of ratings, it is possible that V_j differs from θ_j significantly. The interpretation of the latent topics in such case

[2]RMR is symmetrical in that the topic distribution θ can also be user specific. We found, however, that item specific θ performs better in practice.



(a) LDA (b) CTR (c) HFT (d) RMR

Figure 2: Graphical Models

might be distorted. CTR model was proposed to recommend scientific articles to potential readers, which is a one-class collaborative filtering task [7, 20]. It adopt the strategy to set different c for observed ratings ($X_{i,j} = 1$) and unobserved ratings ($X_{i,j} = 0$). In our setting, we try to match only the observed ratings with the given rating scales (for example, 1 to 5). On the other hand, HFT adopts a fixed transformation function to map one-to-one between θ_j and V_j. In Figure 2(c), we use a dashed line to represent this relationship. The HFT model is effectively a matrix factorization model with the item feature regularization replaced by the corpus likelihood. This fixed transformation function is difficult to select and restrictive in modeling capability.

RMR adopt a mixture of Gaussian to model the ratings. The mixture proportion is assumed to have the same distribution as the topic distribution. Thus when there are few ratings for an item, the text review can still allow us to learn the topic distribution θ accurately. We avoid the difficult choice of the transformation function in HFT and retain the interpretability of the latent topics.

3.3 Collapsed Gibbs Sampler

To develop a Gibbs sampler for RMR, we need to specify the conditional probability of the hidden variables z and f, which are the hidden topics associated with the observed words w and observed ratings x, $P(\mathbf{z}, \mathbf{f}|\mathbf{w}, \mathbf{x})$. This conditional probability does not have a closed form and is difficult to sample directly. The collapsed Gibbs sampler runs a Markov chain that uses the full conditional in order to simulate it. In our case, we need to specify the following two conditional probabilities

$$P(z_i = j|\mathbf{z}_{\neg\mathbf{i}}, \mathbf{w}, \mathbf{f}, \mathbf{x}), \quad P(f_i = j|\mathbf{z}, \mathbf{w}, \mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}). \tag{3}$$

We will briefly derive the expression for the second probability in Eq. (3). Using Bayes' theorem and the conditional independence, we obtain

$$\begin{aligned} &P(f_i = j|\mathbf{z}, \mathbf{w}, \mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}) \\ &\propto P(x_i|f_i = j, \mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}_{\neg\mathbf{i}}) P(f_i = j|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{z}) \end{aligned} \tag{4}$$

Now we will derive the expression for the two terms in Eq. (4). Let the index i denote the rating assigned by user u to item v.

$$\begin{aligned} &P(x_i|f_i = j, \mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}_{\neg\mathbf{i}}) \\ &= \int_{\mu_{u,j}} P(x_i, |\mu_{u,j}, f_i = j) P(\mu_{u,j}|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}_{\neg\mathbf{i}}) \mathrm{d}\mu_{u,j} \end{aligned} \tag{5}$$

The second term in Eq. (5) is a Gaussian distribution, because

$$P(\mu_{u,j}|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}_{\neg\mathbf{i}}) \propto P(\mathbf{x}_{\neg\mathbf{i}}|\mu_{u,j}, \mathbf{f}_{\neg\mathbf{i}}) P(\mu_{u,j}). \tag{6}$$

Since $P(\mu_{u,j})$ is Gaussian $\mathcal{N}(\mu_0, \sigma_0^2)$ and conjugate to $P(\mathbf{x}_{\neg\mathbf{i}}|\mu_{u,j}, \mathbf{f}_{\neg\mathbf{i}})$, the posterior distribution $P(\mu_{u,j}|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}_{\neg\mathbf{i}})$ will be Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$, where

$$\sigma_i^2 = \frac{1}{\sigma_0^2} + \frac{|x^j_{u,(\cdot)}|}{\sigma^2}, \tag{7}$$

$$\mu_i = (\sigma_i^2)^{-1} \left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_m x^j_{u,m}}{\sigma^2}\right). \tag{8}$$

The predictive posterior in Eq. (5) is Gaussian $\mathcal{N}(\mu_i, \sigma_i^2 + \sigma_0^2)$ [19]. Similarly, the expression for the second term in Eq. (4) is

$$P(f_i = j|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{z}) \propto \int_{\theta_v} P(f_i = j|\theta_v) P(\theta_v|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{z}) \mathrm{d}\theta_v, \tag{9}$$

of which the second term is

$$P(\theta_v|\mathbf{f}_{\neg\mathbf{i}}, \mathbf{z}) \propto P(\mathbf{f}_{\neg\mathbf{i}}|\theta_v) P(\mathbf{z}|\theta_v) P(\theta_v). \tag{10}$$

Again, since $P(\theta_v)$ is Dirichlet(α) and conjugate to $P(\mathbf{f}_{\neg\mathbf{i}}|\theta_v)$ and $P(\mathbf{z}|\theta_v)$, the posterior is also a Dirichlet distribution and the posterior predictive of Eq. (9) is

$$P(f_i = j|\mathbf{z}, \mathbf{f}_{\neg\mathbf{i}}) = \frac{n^v_{f,\neg i,j} + n^v_{z,j} + \alpha}{n^v_{f,\neg i,(\cdot)} + n^v_{z,(\cdot)} + K\alpha}. \tag{11}$$

Combine the result in Eq. (5) and Eq. (9), we get the expression for the full conditional for the first probability in Eq. (3)

$$\begin{aligned} &P(f_i = j|\mathbf{z}, \mathbf{w}, \mathbf{f}_{\neg\mathbf{i}}, \mathbf{x}) \\ &\propto \mathcal{N}(x_i|\mu_i, \sigma_i^2 + \sigma_0^2) \frac{n^v_{f,\neg i,j} + n^v_{z,j} + \alpha}{n^v_{f,\neg i,(\cdot)} + n^v_{z,(\cdot)} + K\alpha}, \end{aligned} \tag{12}$$

and by employing a similar procedure, we get the expression for the first probability in Eq. (3)

$$\begin{aligned} &P(z_i = j|\mathbf{z}_{\neg\mathbf{i}}, \mathbf{w}, \mathbf{f}, \mathbf{x}) \\ &\propto \frac{n^{w_i}_{\neg i,j} + \beta}{n^{(\cdot)}_{\neg i,j} + |V|\beta} \frac{n^v_{z,\neg i,j} + n^v_{f,j} + \alpha}{n^v_{z,\neg i,(\cdot)} + n^v_{f,(\cdot)} + K\alpha}. \end{aligned} \tag{13}$$

We summarize the notations used in the derivation process in Table 1. Note that we omit the $\neg i$ subscription in some of the notations to save space. With this notation, it means that when counting

Symbol	Description
$\lvert x^j_{u,(\cdot)}\rvert$	# of ratings assigned by u with topic j
$x^j_{u,m}$	value of rating assigned by u to m with topic j
$n^v_{z,j}$	# of z of item v assigned to topic j
$n^v_{z,(\cdot)}$	total # of review words v received
$n^v_{f,j}$	# of f of item v assigned to topic j
$n^v_{f,(\cdot)}$	total # of ratings item v received
$n^{w_i}_j$	# of word w_i assigned to topic j
$n^{(\cdot)}_j$	total # of words assigned to topic j

Table 1: Notations

the respective values, we exclude current word w_i or current rating x_i.

3.3.1 Readout Parameters

Once the sampling process is finished, we can readily readout the model parameters

$$\theta_{v,j} = \frac{n^v_{z,j} + n^v_{f,j} + \alpha}{n^v_{z,(\cdot)} + n^v_{z,(\cdot)} + K\alpha}, \psi_{j,w_i} = \frac{n^{w_i}_j + \beta}{n^{(\cdot)}_j + |V|\beta}, \quad (14)$$

and

$$\mu_{u,j} = \left(\frac{1}{\sigma_0^2} + \frac{|x^j_{u,(\cdot)}|}{\sigma^2}\right)^{-1} \left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_m x^j_{u,m}}{\sigma^2}\right). \quad (15)$$

The notations denote the same meaning as is in Table 1, except that the counters now count all effective samples.

3.3.2 Time and Space Complexity of the Sampler

Our collapsed Gibbs sampler mainly use the following counters to keep track of current states: counter $n^v_{z,j}$ of size $M \times K$, counter $n^v_{f,j}$ of size $M \times K$, counter $n^w_{\neg i,j}$ of size $V \times K$, counter $\sum_m x^j_{n,m}$ of size $N \times K$ and counter $|x^j_{n,(\cdot)}|$ of size $N \times K$. Other than the above listed counters, there are summation counters that are one order smaller than the above counters and thus negligible. So the total Space complexity is $O((M + N + V) \times K)$.

To sample z or f conditioned on everything else, we only need to calculate the conditional probabilities in Eq. (3). This operation requires $O(K)$ operations. Given a fixed K, our sampler scales *linearly* with the length of the observed review text and the number received ratings. Usually the number of latent topics K is small, making our sampler scales up well. Note that training a RMR model is faster than training a HFT model [16]. This is because the later one requires an additional step to learn the feature vectors for all the users.

3.4 Prediction

In order to make a prediction that user u assigns to item v, we compute the expected value

$$x_{u,v} = \sum_k \theta_{v,k}\mu_{v,k}. \quad (16)$$

In practice, we compute the empirical global mean g, user bias b_u and item bias b_v for all users and all items from the training set. We feed the $x'_{u,v} = x_{u,v} - g - b_u - b_v$ to the sampler and when making predictions, add g, b_u, and b_v back.

4. EXPERIMENTS

We conduct an empirical study of RMR and various baseline models to show the following facts:

1. Our model leads to significant improvement on prediction accuracy across various categories of items over several strong baseline models.

2. Our model learns latent topic dimensions that are clearly interpretable.

3. Our model performs better in datasets which are extremely sparse, which resembles the cold-start settings.

4.1 Dataset

We use the Amazon Review dataset collected by [16]. This dataset is a collection of 27 datasets corresponding to various types of items that are available on Amazon[3]. This is the largest rating dataset with text reviews publicly available, to the best of our knowledge. We show the statistics of the datasets in Table 2. Refer to Table 2; there are two facts that are evident immediately. First, the datasets are extremely sparse. The sparseness would clearly deteriorate the performance of most existing recommender systems which only model the ratings. Secondly, a review contains 116.87 words on average across all categories. As will be apparent in the results shown later, these review texts are key to model the ratings accurately.

4.2 Baseline Methods

We compare our model with four baseline models MF, LDAMF, CTR and HFT.

- **MF** This is the standard matrix factorization model as is described in [25]. We ignore the review texts completely and model the ratings only. This is typically a very strong baseline model in collaborative filtering [10, 23].

- **LDAMF** This baseline model is proposed in [16]. This baseline model tries to harness the information in the review text by fitting an LDA model on the review text and then treat the learnt topic distribution on items (or users) as the latent factors in matrix factorization models. By holding the latent factors for items (or users) fixed, the latent factors for users (or items) are learnt by gradient descent methods.

- **CTR** This is the state-of-the-art method that recommends scientific articles to potential interested readers [31]. The CTR model solves the one-class collaborative filtering problem by using different precision parameter c. In our setting, we use it to match the observed integer ratings using the same precision c. We employ LDA-C [3] to pre-train the model. Note that CTR utilizes both ratings and reviews information.

- **HFT** This is the state-of-the-art method that combines reviews with ratings [16]. HFT models the ratings using a matrix factorization model with an exponential transformation function to link the stochastic topic distribution in modeling the review text and the latent vector in modeling the ratings. The topic distribution can be modeled on either users or items. On most datasets, the item specific topic distribution produces more accurate predictions. We report the results whichever are more accurate.

[3]The statistics of category Baby we calculated differs from the description provided on the webpage and we exclude it from consideration.

Dataset	#users	#items	#review	#words	words/review	reviews/item
Arts	24,071	4,211	27,980	2,006,874	71.73	6.64
Jewelry	40,594	18,794	58,621	3,100,948	52.90	3.12
Industrial Scientific	29,590	22,622	13,7042	6,920,151	50.50	6.06
Watches	62,041	10,318	68,356	5,436,671	79.53	6.62
Cell Phones and Accessories	68,041	7,438	78,930	7,567,961	95.88	10.61
Musical Instruments	67,007	14,182	85,405	7,442,294	87.14	6.02
Software	68,464	11,234	95,084	11,012,882	115.82	8.46
Gourmet Foods	112,544	23,476	154,635	10,542,984	68.18	6.59
Office Products	110,472	14,224	138,084	11,206,338	81.16	9.71
Automotive	133,256	47,577	188,728	13,249,641	70.21	3.97
Patio	166,832	19,531	206,250	17,290,881	83.83	10.56
Pet Supplies	160,496	17,523	217,170	18,684,153	86.03	12.39
Beauty	167,725	29,004	252,056	17,889,577	70.97	8.69
Shoes	73,590	48,410	389,877	23,604,059	60.54	8.05
Kindle Store	116,191	4,372	160,793	21,533,201	133.92	36.78
Clothing and Accessories	128,794	66,370	581,933	34,267,151	58.89	8.77
Health	311,636	39,539	428,781	33,277,423	77.61	10.84
Toys and Games	290,713	53,600	435,996	35,034,001	80.35	8.13
Tools and Home Improvement	283,514	51,004	409,499	34,591,409	84.47	8.03
Sports and Outdoors	329,232	68,293	510,991	38,898,738	76.12	7.48
Video Games	228,570	21,025	463,669	55,532,148	119.77	22.05
Home and Kitchen	644,509	79,006	991,794	81,923,017	82.60	12.55
Amazon Instant Video	312,930	22,204	717,651	88,958,349	123.96	32.32
Electronics	811,034	82,067	1,241,778	124,064,510	99.91	15.13
Music	1,134,684	556,814	6,396,350	774,791,468	121.13	11.49
Movies and TV	1,224,267	212,836	7,850,072	997,261,969	127.04	36.88
Books	2,588,991	929,264	12,886,488	1,613,603,531	125.22	13.87
All categories	6,643,669	2,441,053	34,686,880	4,053,795,667	116.87	14.21

Table 2: Statistics of the datasets

4.3 Evaluation

We use Mean Squared Error (MSE) to evaluate various models. For each of the dataset, we randomly select 80% as training set up to 2 million reviews. The remaining reviews are split evenly into validation set and testing set. The initial latent variables z and f are uniformly randomly assigned. We run 2500 iterations with a thinning of 50 iterations to get samples and MSE readout. We report the MSE of the testing set which has the lowest MSE on the validation set. The training of the baseline methods MF, LDAMF, CTR and HFT follow the same routine described in [16]. We use $K = 5$ for all models. We set hyperparameters[4] $\alpha = 0.1$, $\beta = 0.02$, $\mu_0 = 0$, $\sigma_0^2 = 1$ and we use the empirical variance of x as σ^2. In practice, the time required to train the RMR model is about half the time spend on training the HFT model on the same machine.

4.4 Rating Prediction

Shown in Table 3 are the MSE results. The best MSE of each dataset is in bold. We listed the performance of various models on the datasets and the average improvement. The standard deviations of MSE results are shown in parenthesis. Out of the 27 datasets, RMR performs the best on 19 datasets among all considered methods.

Compared with matrix factorization (MF column in Table 3), RMR performs better on 26 out of the 27 datasets with an average improvement on MSE of nearly 8%. Matrix factorization method usually performs well in practice [10, 23] and is a strong baseline method. However, as is shown in our case, in datasets which are extremely sparse, MF is unable to learn an accurate representation of users/items and thus under-performs other methods which take the review text into consideration. However, in the datasets such as Music, Movies and TV and Books, which are relatively denser compared with other datasets; the MF method still performs very well.

The baseline method LDAMF, which was proposed as a baseline method in [16], is probably the simplest model that combines review text and ratings. This baseline method takes the item topic distribution produced by LDA as the feature vectors for the items and then learns the user feature vectors by fitting the observed ratings with item features fixed. The feature vectors of items are learnt using *only* the reviews, which might be sub-optimal to fit the rating data. The expressiveness is thus restricted and we think this restriction caused the nearly 8% improvement produced by RMR.

Compared with CTR, which take the full advantage of the combined information of both the reviews and ratings, our proposed model still leads to an average improvement of 3.28% and performs better on 25 out of the 27 datasets. Similar to LDAMF, CTR takes the item topic distribution produced by LDA as the initial item features. However unlike LDAMF, during the training period, CTR alters both the user features and item features to fit the ratings. The regularization parameter λ_V controls how much the item features can deviate from the item topic distribution vectors. It performs better due to the more flexible modeling capability. However, the CTR does not perform as well as RMR in the extremely sparse datasets such as Arts and Jewelry. We observe that during the experiment, CTR can learn a model that fits the data with a small training error. But the generalization of the learnt model to the unobserved rating is not as good. Note that we report the performance of CTR on the test set by setting λ_V and λ_U to the value which gives best performance on validation set. So the issue of under-regularization is minimized. The performance of CTR on the relatively dense datasets is very competitive.

Compared with HFT, another recommendation method that takes review text into consideration, RMR is still able to improve the performance by 1.22% on average and performs better or equally well in 21 out of 27 datasets. As discussed in previous sections, we think

[4]We searched through the parameters linearly and reported hyperparameters which performed the best.

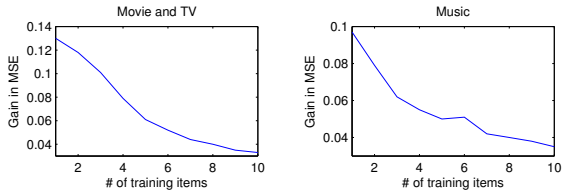


Figure 3: Gain in MSE for user with limited training data

the fixed one-to-one mapping between the item topic distribution and item feature vector impose restrictions of the expressiveness of HFT and allow RMR to out-perform it. Due to large size of the datasets, the improvements reported are significant at 1% level.

We consider the improvements of RMR over CTR (3.28%) and HFT (1.22%) significant because both of these two baselines are full-fledged models that take both the ratings and reviews into consideration. Also, these improvements are verified on 27 real-life datasets. In a real system where recommendation plays a central role, e.g. Amazon, Netflix, these improvements could lead to better revenue and profit.

4.5 Cold-start Setting

An interesting phenomenon we found in the results is that the improvement of RMR over the traditional collaborative filtering methods (matrix factorization) is more significant for datasets that are sparse. For classes such as Arts, Industrial Scientific, RMR show substantial improvement. In such cases, the number of ratings is too scarce to model the items and users adequately. The text in the review associated with the ratings come as rescue, which allow our model to learn a more accurate topic distribution. Whereas for classes such as Music, Movies and Books, which are the largest 3 datasets with larger reviews per user and reviews per item, the traditional methods tend to produce accurate predictions. We further verify this finding by comparing the performance of RMR with MF on users with limited training data. Shown in Figure 3 is the gain of RMR compared with MF for users with limited training items. We show the result on two datasets due to space limit and the phenomenon repeats across all the datasets. As we can see, our model gains the most when the user has few training items. The performance gain starts to decrease with the number of training items available for each user. This further demonstrates that RMR is valuable for the cold-start settings.

4.6 Interpretability of Topics

Apart from being more accurate at prediction, another advantage of RMR is that it learns interpretable latent topics. We show two examples of the top words in each topic learnt in RMR in Table 4 and Table 5. Table 4 shows the top words for topics learnt with *software* dataset. Note that Roxio is software for burning DVDs and Quicken is personal financial software. Leopard and Tiger are the code name of Mac OS X and Parallels is a popular virtual machine on OS X. The fourth topic is about the company Microsoft and its products and the last topic is related to Linux. Table 5 shows the top words for topics learnt with Movie and TV dataset. The first topic is dedicated to workout related videos. The second topic contains commonly used words to describe TV series. Batman, Matrix trilogy, Alien and Harry Potter are either science fiction, adventure or fantasy movies. Godzilla is a disaster thriller and Hitchcock is

roxio	quicken	leopard	office	suse
contacted	son	os	excel	accounts
perfect	pick	parallels	2007	2004
burning	given	apple	student	nav
dvds	spanish	turbo	activation	federal
care	starting	tiger	microsoft	symantec

Table 4: Top words for topics in Software

workout	season	batman	disney	godzilla
yoga	match	effects	christmas	hitchcock
workouts	episodes	alien	animation	kidman
videos	seasons	harry	kids	murder
exercises	vs	matrix	shrek	densel
cardio	episode	edition	animated	nicole

Table 5: Top words for topics in Movie and TV

a famous director of psychological thrillers. Nicole Kidman is the leading actress of the classic thriller "Eyes Wide Shut".

Clearly these interpretable topics would help us understand items and users better. For items, the top topic words can be employed as extended tags attached to the item and may improve the prediction accuracy in a tag-aware recommender system [5]. We may also gain better understanding of items by analyzing the topic distribution similarities. For users, once obtaining the topic preferences, we can recommend "cold" items which have few or no ratings to the users with confidence. For example, if we know that a user tends to rate high for topic three and five in Table 5, we can confidently recommend the movie "Interstellar" (a Sci-Fi Thriller movie) even if this movie is not being shown yet. Our prior knowledge of items therefore can help alleviate the cold-start problem.

5. CONCLUSION

In this paper, we propose a model that combines content-based filtering with collaborative filtering seamlessly. By exploiting the information in both ratings and reviews, we are able to improve the prediction accuracy significantly across various classes of datasets over existing strong baseline methods, especially under the cold-start settings where the data are extremely sparse. We develop an efficient collapsed Gibbs sampler for learning the model parameters. Our model also learns topics that are interpretable, enabling us to exploit prior knowledge to alleviate the cold start problem. We plan to explore RMR's ability in discovering user communities and new genres in future work.

Acknowledgement

The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340401 and No. 2014CB340405), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413212 and CUHK 415113), and Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036).

6. REFERENCES

[1] D. Agarwal and B.-C. Chen. flda: matrix factorization through latent dirichlet allocation. In *WSDM*, pages 91–100, 2010.

[2] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *ICML*, 2004.

	a	b	c	d	e	Improvement of RMR versus		
Dataset	MF	LDAMF	CTR	HFT	RMR	min(a,b)	c	d
Arts	1.565 (0.04)	1.575 (0.04)	1.471 (0.04)	1.390 (0.04)	**1.371 (0.04)**	14.15%	7.29%	1.39%
Jewelry	1.257 (0.03)	1.279 (0.03)	1.206 (0.03)	1.177 (0.02)	**1.160 (0.02)**	8.36%	3.97%	1.47%
Industrial Scientific	0.461 (0.02)	0.462 (0.02)	0.382 (0.02)	**0.359 (0.02)**	0.362 (0.02)	27.35%	5.52%	-0.83%
Watches	1.535 (0.03)	1.518 (0.03)	1.491 (0.03)	1.488 (0.03)	**1.458 (0.02)**	4.12%	2.26%	2.06%
Cell Phones and Accessories	2.230 (0.04)	2.308 (0.04)	2.177 (0.04)	2.135 (0.03)	**2.085 (0.03)**	6.95%	4.41%	2.40%
Musical Instruments	1.506 (0.02)	1.520 (0.02)	1.422 (0.02)	1.395 (0.02)	**1.374 (0.02)**	9.61%	3.49%	1.53%
Software	2.409 (0.02)	2.214 (0.02)	2.254 (0.02)	2.219 (0.02)	**2.173 (0.02)**	1.89%	3.73%	2.12%
Gourmet Foods	1.515 (0.01)	1.491 (0.01)	1.482 (0.01)	**1.457 (0.01)**	1.465 (0.01)	1.77%	1.16%	-0.55%
Office Products	1.814 (0.01)	1.796 (0.01)	1.733 (0.01)	1.669 (0.01)	**1.638 (0.01)**	9.65%	5.80%	1.89%
Automotive	1.570 (0.01)	1.585 (0.01)	1.492 (0.01)	1.432 (0.01)	**1.403 (0.01)**	11.90%	6.34%	2.07%
Patio	1.771 (0.01)	1.793 (0.01)	1.720 (0.01)	1.698 (0.01)	**1.669 (0.01)**	6.11%	3.06%	1.74%
Pet Supplies	1.700 (0.01)	1.700 (0.01)	1.613 (0.01)	1.583 (0.01)	**1.562 (0.01)**	8.83%	3.27%	1.34%
Beauty	1.399 (0.01)	1.414 (0.01)	1.361 (0.01)	1.358 (0.01)	**1.334 (0.01)**	4.87%	2.02%	1.80%
Shoes	0.305 (0.00)	0.335 (0.00)	0.271 (0.00)	**0.247 (0.00)**	0.251 (0.00)	21.51%	7.97%	-1.59%
Kindle Store	1.553 (0.01)	1.561 (0.01)	1.457 (0.01)	1.437 (0.01)	**1.412 (0.01)**	9.99%	3.19%	1.77%
Clothing and Accessories	0.393 (0.00)	0.406 (0.00)	0.355 (0.00)	0.349 (0.00)	**0.336 (0.00)**	16.96%	5.65%	3.87%
Health	1.615 (0.01)	1.608 (0.01)	1.552 (0.01)	1.538 (0.01)	**1.512 (0.01)**	6.35%	2.65%	1.72%
Toys and Games	1.467 (0.01)	1.395 (0.01)	1.389 (0.01)	**1.370 (0.01)**	1.372 (0.01)	1.68%	1.24%	-0.15%
Tools and Home Improvement	1.600 (0.01)	1.610 (0.01)	1.513 (0.01)	1.510 (0.01)	**1.491 (0.01)**	7.31%	1.48%	1.27%
Sports and Outdoors	1.219 (0.01)	1.223 (0.01)	1.150 (0.01)	1.138 (0.01)	**1.129 (0.01)**	7.97%	1.86%	0.80%
Video Games	1.610 (0.01)	1.608 (0.01)	1.572 (0.01)	1.528 (0.01)	**1.510 (0.01)**	6.49%	4.11%	1.19%
Home and Kitchen	1.628 (0.05)	1.610 (0.05)	1.577 (0.05)	1.531 (0.04)	**1.501 (0.04)**	7.26%	5.06%	2.00%
Amazon Instant Video	1.330 (0.01)	1.328 (0.01)	1.291 (0.01)	**1.260 (0.01)**	1.270 (0.01)	4.57%	1.65%	-0.79%
Electronics	1.828 (0.00)	1.823 (0.00)	1.764 (0.00)	**1.722 (0.00)**	**1.722 (0.00)**	5.87%	2.44%	0.00%
Music	**0.956 (0.00)**	0.958 (0.00)	0.959 (0.00)	0.980 (0.00)	0.959 (0.00)	-0.31%	0.00%	2.19%
Movies and TV	1.119 (0.00)	1.117 (0.00)	**1.114 (0.00)**	1.119 (0.00)	1.120 (0.00)	-0.27%	-0.54%	-0.09%
Books	1.107 (0.00)	1.109 (0.00)	**1.106 (0.00)**	1.138 (0.00)	1.113 (0.00)	-0.54%	-0.63%	2.25%
Average on all datasets						7.79%	3.28%	1.22%

Table 3: MSE results of various models

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, 2009.

[5] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social media recommendation based on people and tags. In *SIGIR*, pages 194–201, 2010.

[6] X. Hu, J. Tang, H. Gao, and H. Liu. Unsupervised sentiment analysis with emotional signals. In *WWW*, pages 607–618, 2013.

[7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[8] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, 2011.

[9] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[10] Y. Koren and J. Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *RecSys*, pages 117–124, 2011.

[11] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, January 2003.

[12] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. 2012.

[13] N. N. Liu, X. Meng, C. Liu, and Q. Yang. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *RecSys*, pages 37–44, 2011.

[14] H. Ma, I. King, and M. R. Lyu. Mining web graphs for recommendations. *IEEE Trans. Knowl. Data Eng.*, 24(6):1051–1064, 2012.

[15] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.

[16] J. J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, pages 165–172, 2013.

[17] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[18] S. Moghaddam and M. Ester. Aspect-based opinion mining from product reviews. In *SIGIR*, page 1184, 2012.

[19] K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, University of British Columbia, 2007.

[20] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511, 2008.

[21] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341, 2007.

[22] S. Purushotham and Y. Liu. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, 2012.

[23] S. Rendle. Factorization machines with libfm. *ACM TIST*, 3(3):57, 2012.

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *CoRR*, abs/1205.2618, 2012.

[25] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.

[26] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887, 2008.

[27] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[28] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260, 2002.

[29] A. Sharma and D. Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *WWW*, pages 1133–1144, 2013.

[30] I. Titov and R. T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, pages 308–316, 2008.

[31] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456, 2011.

[32] L. Zhang, D. Agarwal, and B.-C. Chen. Generalizing matrix factorization through flexible regression priors. In *RecSys*, pages 13–20, 2011.