Using Critic Reviews to Boost New Item Recommendation

Completed Research Paper

Xiaoying Xu
Department of Information Systems
National University of Singapore
15 Computing Drive, Singapore, 117418
xu.xiaoying@comp.nus.edu.sg

Kaushik Dutta
Department of Information Systems
National University of Singapore
15 Computing Drive, Singapore, 117418
duttak@nus.edu.sg

Anindya Datta
Department of Information Systems
National University of Singapore
15 Computing Drive, Singapore, 117418
datta@comp.nus.edu.sg

Abstract

The rapid development of technology promotes the vast expansion of new items in many domains of consumer products. Problem occurs when the new items are continuously added but cannot get reached by the consumers. Many existing recommender systems work well only for well-known items with sufficient ratings but fail to discover new items, and content-based approaches suffer from insufficient item features. In this paper, we show that critic reviews of the items can be used to boost new item recommendation. We propose a scalable framework that incorporates the topics inferred from the critic reviews into the recommendation process by employing topic modeling and non-negative matrix factorization. The results of our experiment show that our proposed method is able to generate high quality new item recommendations which are not supported by many state-of-the-art methods, and also outperforms the state-of-the-art methods in recommending existing items.

Keywords: Recommender system, new item, critic reviews, topic model, PLDA
Introduction

The fast development of technology continuously increases the productivity, reduces the production cost, and shortens the production cycle of new products. Facilitated by the rapid emergence of easy-to-use tools and software, individuals now can create new items by themselves at a relatively low cost, and a plenty of online publishing platforms have offered a wide range of opportunities for them to upload and release their works. For example, video production that used to be regarded as high-cost task and that only could be undertaken by a small number of powerful companies before, now becomes very common among individuals who can freely release their videos through online video platforms like YouTube\(^1\). YouTube reported that they had more than 800 million users uploading over an hour of video per second. Clearly, the barrier of entry for individuals to create new items has become much lower than ever before. As a consequence, in most domains of consumer products studied, new items are added regularly at a very high speed never seen before. For example, according to (Datta et al. 2012), on average 100 new movies, 250 new books and up to 15,000 new mobile apps are released per week. These huge number of new items constitute the “long tail” (Anderson and Andersson 2007) of item distribution, which can hardly get reached by consumers without any mechanism that effectively supports new item discovery. In a recent development, Recommender Systems (RSs) have shown promise of helping consumers make good choices among overwhelming number of alternative items by providing personalized recommendations. Unfortunately, most of the recommendation techniques work well only if the items are already well-known (i.e. the items have been purchased or rated by many users). They lack the ability to discover and recommend new items since the user ratings required by these techniques are extremely sparse or totally unavailable among the new items. Problem occurs when new items are continuously added but cannot get recommended. Motivated by the huge demand for new item discovery, in this paper, we propose a scalable framework that incorporates external source to boost high quality recommendations of new items.

An intuitive solution to the new item recommendation is to adopt the content-based approaches that typically match user preference data with item attribute information to help bridge the gap between the existing items and the new ones. Nevertheless, there are a number of inadequacies identified in existing content-based approaches. First, some existing content-based methods use the descriptive attributes of items, e.g. the genre, the directors and the actors of the movies, to infer users’ preference. However, research found that limited number of descriptive attributes assigned to items could be insufficient to define distinguishing features of items which turn out to be necessary for the elicitation of users’ taste aspects (Lops et al. 2011). For example, a user may prefer dramas about school life but dislike dramas about racial discrimination. If genre is used as an indicator of users’ preference, it fails to differentiate these two detail aspects of user taste. Moreover, some users’ taste aspects may not belong to a specific descriptive attribute. For example, a user may prefer movies about family that can appear in any genre. To address this problem, more item information should be incorporated to define the user taste aspects. Second, some content-based methods also use the textual description of the items to represent the items and capture the user taste. Such methods usually represent the item characteristics and users’ taste aspects at the word level, which may cause the over-specification problem. For example, a user may prefer family movies in whose textual descriptions the word “mother” may appear frequently, but it does not mean that this user must like all the movies whose textual descriptions contain “mother”. Higher level and more abstract representations of the textual contents are required by these methods. Third, most of the content-based methods cannot fully utilize the user ratings. They usually convert the user ratings into binary attributes (i.e. like or dislike) and then regard the recommendation procedure as a classification problem (Bouza et al. 2008; Christakou et al. 2007; Xu and Araki 2006). However, if the scale of user rating is 1 to 10, a rating of 8 and a rating of 10 may both indicate that this user likes the item, but the extents of preference are different. The scale of the user ratings contains useful information of user preference but it is discarded by these methods. A proper method to fully utilize the user ratings and item information would be helpful in improving the recommendation quality of the existing content-based approaches.

To address these problems, we require much more information about the new items other than a limited number of item attributes to represent the characteristics of the new items and to define the detail aspects

\(^1\) http://www.youtube.com
of user taste. We started off exploring other possible external contents that could be incorporated and noticed that the critic reviews of the corresponding items are available in abundance even before the new items are released to the public. For example, 22 high quality critic review articles of a recent movie Mud can be found in Rotten Tomatoes\(^2\), which is a movie critic review aggregator, one week before its release. These review articles may cover all possible aspects of the items and therefore provide us with a rich source of new item information. We collect the critic reviews from external source and incorporate them in our recommending process. By representing the critic reviews at the topic level, we are able to tell the characteristics of the items at a high level without any user ratings. For example, given the critic reviews of the movie \textit{The Graduate}, we can infer that this movie is about school life, and we can also know the movie \textit{American History X} is about racial discrimination through its critic reviews. The topics of the items’ critic reviews are good indicators of user taste since they cover much more features of the items than the item attributes and they are at the higher and more abstract level without the over-specification problem. We use the topics of the critic reviews from existing items to define the taste aspects of the users, and utilize the user ratings to estimate the extent to which a user likes a particular topic. When the new items are added, we collect their critic reviews, and infer their topic distributions. Then the recommendation can be performed by matching the users’ topic preference with the topic distributions of the new items.

To the best of our knowledge, this is the first work to incorporate external critic reviews in recommendation. We prove that critic reviews can be used to boost new item recommendation. The results of our experiment conducted in a real world data set show that our method is effective and can not only generate high quality new item recommendations which are not supported by many state-of-the-art methods, but also outperform the state-of-the-art methods when recommending existing items especially in rating-sparse settings. Specifically, our method reduces the prediction errors of the state-of-the-art method using item typology based on item keywords by 5.78\% and improves the ranking accuracy of the state-of-the-art method by 12.91\% in rating-sparse settings.

The rest of this paper is organized as follows. First we introduce the background and review the related work. Then we present our proposed recommendation framework including the intuition and the detail description of each component. The remainder of the paper then presents the experiment and results, and finally, we conclude by summarizing the paper, including the contribution, limitation and implication for future research.

\section*{Background and Related Work}

Our method is a content-based approach using the external critic reviews of items to boost new item recommendation. We employ advanced topic modeling approach, i.e. Partially Labeled Dirichlet Allocation (PLDA) to represent the critic reviews at the topic level. We also adapt Non-negative Matrix Factorization (NMF) to fully utilize the rating and content information. The background and related work will be introduced in the following.

\section*{Content-based Recommendation}

Content-based recommendation aims at recommending items that are similar to what the target user has liked in the past. A typical content-based RS constructs a profile which is a structured representation of interests for every user by analyzing the description of items previously rated by this user, and the recommendation process is to match up the user profile against the attributes of new items (Pazzani and Billsus 2007). The nature of content-based approach enables new item recommendation since it does not require any user preference data of the new items.

Content-based RSs working in the domains of consumer products (books, movies, mobile apps, etc.) usually use the well-structured attributes to represent the items. For example, the genre, the directors and the actors of the movies have been commonly used in movie RSs (Gantner et al. 2010; Maneeroj and Takasu 2009; Manzato 2012). Such systems are limited by the insufficient number of descriptive

\footnote{http://www.rottentomatoes.com}
attributes associated with the items. Some studies have also proposed to use external source in addition to internal information to overcome the limitation. For example, Katz, et al. (2011) proposed to map items to the Wikipedia pages and compute the similarities between items based on the page text, the belonging categories, and links between Wikipedia articles. However, no reported work has been found to use the external critic reviews on corresponding items, which are incorporated in our approach.

Different from the well-structured data, critic reviews are presented in the form of free text. Text processing is required to construct the item representation. In some content-based RSs, the descriptions of items or the items themself are textual contents, e.g. Web pages, news articles, or product descriptions. Most of such systems use Vector Space Model (VSM) (Salton et al. 1975) to represent items. Specifically, each item is represented as a sparse vector of TF-IDF term weights (Salton and Buckley 1988) that indicate the degree of association between the terms and the item. For example, Fab (Balabanović and Shoham 1997) is a web page RS that represents a web page as a vector of top 100 words with highest weights. A personalized news RS YourNews (Ahn et al. 2007) uses 8 term vectors to represent the news viewed by a user, each of which is corresponding to one topic. Spaeth and Desmarais (2013) performed people-people recommendation by computing text similarities of the expert profiles represented as TF-IDF term vectors. Despite of its popularity in content-based RSs, VSM fails to discover the underlying topics that help reduce the dimension of the text contents and that are essential for recommendation. This shortcoming is addressed in our proposed method by employing PLDA that will be introduced later.

In content-based RSs, user profile learner is a core component. Many existing methods have regarded the user preference as a binary attribute (i.e. like or dislike) and therefore, the recommendation problem can be converted into a classification problem. A series of classification learners have been applied to learn the user profiles, including Decision Tree (Bouza et al. 2008), Bayesian classifier (Gutta et al. 2000), SVM (Xu and Araki 2006), Neural Network (Christakou et al. 2007) etc. These methods are criticized due to their high complexity and poor interpretability. They also fail to utilize the user ratings. In our method, we proposed a NMF approach to address these problems.

**Partially Labeled Dirichlet Allocation**

Probabilistic topic models are a series of statistical models applied to discover the hidden thematic structure of documents in text corpus. Among the topic models, Latent Dirichlet Allocation (LDA) (Blei et al. 2003) is the simplest one and it has gained popularity among theoreticians and practitioners. LDA is a generative probabilistic model that applies hierarchical Bayesian analysis to discover the semantic structure in a text corpus. The basic idea of LDA is to represent a document as a multinomial distribution over latent topics, each of which is characterized by a distribution over words. LDA is a completely unsupervised learning model. The unsupervised topics generated by LDA have strength in exploring the underlying sub-structure, but it may be difficult to interpret their meaning and they usually do not align with human provide labels. Labeled LDA (Ramage et al. 2009) is a supervised extension of LDA that requires the topics to align with the pre-defined labels assigned to the documents, and it does not assume the existence of any latent topics. Labeled LDA improves the interpretability of the learned topics, but the strong constraint may make it fail to capture the broad patterns in the corpus. In a recent development, a semi-supervised model, i.e. Partially Labeled Dirichlet Allocation (PLDA) (Ramage et al. 2011), has been proposed. PLDA takes full advantage of both supervised and unsupervised approaches. It is able to discover any number of hidden topics under each pre-defined label, and it also has the ability to explore the background latent topics across the whole corpus. There are a few existing works applying LDA in recommendation. For example, a recent study (Cai et al. 2013) proposed a TyCo method which uses LDA to model keywords of movies and then construct item typicality for further recommendation. But no reported work using PLDA in recommendation has been found. The nature of PLDA makes it suitable for our purpose to uncover topics in the critic reviews. The learned topics are then incorporated with user ratings by applying NMF in our method.

**Non-negative Matrix Factorization**

Non-negative Matrix Factorization (NMF) (Lee and Seung 1999) is a powerful dimension reduction tool for non-negative data and has been successfully adopted in many fields such as signal processing and text
mining. Given a non-negative matrix $V \in \mathbb{R}^{m \times n}$ and a specified positive integer $k < \min(m, n)$, NMF seeks for two non-negative matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ so that their product $WH$ approximates the original matrix $V$. The intuition of NMF is to use a linear combination of the basis vectors (i.e. the rows in $W$) and the coefficient vectors (i.e. the columns in $H$) to approximate the input vectors (i.e. the rows in $V$). NMF can be solved as a problem of minimizing the error function, which is typically the square error or Kullback-Leibler divergence, and coordinate descent algorithms (Hsieh and Dhillon 2011; Seung and Lee 2001) are commonly used. In our method, we adapt NMF to our context by redefining the error function and use a simple and effective projected gradient descent approach (Lin 2007) to solve the optimization problem.

**Recommendation Framework**

In this section, we are going to describe our proposed method, starting from intuition behind it, and continuing to show the recommendation framework and the details of its each component. Although our proposed method may be generally applicable in any domain of consumer products, we make it easier to explain our idea and to compare our work with existing ones by choosing a specific example domain. Noticing that movie is the most studied consumer domain in recommendation research, we will present our work upon the movie domain. That is to say, from this point forward, we will present our method as a technique of providing movie recommendations to users.

**Intuition and Overview**

The objective of our method is to predict the users’ preference on their unknown movies, and to recommend those movies with highest predicted ratings to them. In order to predict the target user’s preference on a given movie, we need to know what kinds of movie he has liked in the past, and what kinds of movie the given movie belongs to. A common way to do this is to use the descriptive attributes of the movies such as the genre and the director to define the characteristics of the movie and the users’ taste aspects. For example, if this user has highly rated science fiction movies directed by Spielberg, and the given movie happens to be science fiction and directed by Spielberg, then the predicted rating would be higher.

However, a user’s taste may far beyond the aspects defined by the limited number of descriptive attributes. For example, a user may prefer comedies about school life or dramas about racial discrimination. This leads us to think about if there are other types of data that can be incorporated to cover more features of the movies and to capture more aspects of users’ taste. We notice that a specific kind of movie information, i.e. critic reviews, can be found in many online systems like Rotten Tomatoes, which is widely known as a movie review aggregator, and IMDb\(^3\), which is a popular movie database. Unlike other user generated contents or user preference data that only can be found long after the movie is released to the public, critic review articles are available in abundance even before the release of the movie. For example, 22 high quality critic review articles of a recent movie *Mud* can be found in Rotten Tomatoes even one week before its release. The availability of expert critic reviews fulfills our requirement for the information of new movies, and their contents may cover all possible aspects of the movies. Therefore, we incorporate external expert critic reviews to define movie features.

Since critic reviews are presented in the form of free text, proper text model should be used to represent movies with these textual contents. We apply PLDA to the critic reviews to infer the topics of movies under their genres as well as the topics that are shared by all the genres. For example, a topic under the genre drama may be related to “racial discrimination”, and a general topic may be related to “family” since many movies in different genres may all talk about family. The adaption of NMF allows us to calculate the users’ preference on each topic based on their rating data. Therefore, the rating prediction becomes a problem of estimating to what extent the movie topic distribution matches the user topic preference. Given a new movie with a collection of expert critic reviews, we are able to tell to what extent this movie is self-referring.

---

\[^3\]http://www.imdb.com

---
associated with which topic and new item recommendation can be performed by matching up the movie topic distribution with the user topic preference.

**Proposed Framework**

The overview of our movie recommendation framework is shown in Figure 1. We have designed and implemented four components in the framework to realize our recommendation engine. The four components are: Crawler, Topic Modeler, Profile Learner and Recommender. Among these components, Recommender is the only one running online, while the other three can run offline. Specifically, for the existing movies, we use the Crawler to collect critic reviews from external websites. The contents of critic reviews are then analyzed by the Topic Modeler to uncover the underlying topics for representing the movies. The Profile Learner utilizes the user ratings and the movie topic distributions to learn the user preference. And the Recommender generates personalized movie recommendations by matching up the user preference with the movie features. For new movies, their topic distribution is inferred from the topic model trained by the existing movies, and then are used together with the user profiles learned from the existing movies to generate recommendations. The PLDA employed in the Topic Modeler and the adaption of NMF in Profile Learner make our proposed method distinguishing from other ones, which also contribute to more efficient and higher quality recommendations. More details of each component will be introduced in the ensuing sections of the paper.

![Proposed Framework Diagram](image)

**Crawler**

The main task of Crawler is to collect critic reviews from external websites via Rotten Tomatoes. Rotten Tomatoes aggregates critic reviews from reliable sources with high reputation and offers a list of their
URLs. It also allows users to select reviews from top critics evaluated by their produced quality. Figure 2 is a snapshot that shows the summary of critic reviews for a movie in Rotten Tomatoes. We use the titles and release years of the movies to match the movie information in Rotten Tomatoes via its search API and then get a list of critic review URLs. For each movie, we crawl 20 webpages of critic reviews. To ensure the quality, we primarily use the reviews from top critics. If their number is less than 20, we also use reviews from other critics. Since the reviews are from different websites, the structures of the webpages containing the review contents are different. We need to use a content extractor (Kohlschütter et al. 2010) to extract the review contents from these webpages. We filter out those reviews that are not written in English and those that are too short (less than 100 words). The extracted review contents are then passed to the Topic Modeler.

![Critic Reviews for Titanic](image)

**Figure 2. Critic Reviews in Rotten Tomatoes**

**Topic Modeler**

As input of the Topic Modeler, a movie is represented as a vector of which the elements are the frequencies of words in its critic reviews. The function of Topic Modeler is to construct the topic representation for the movies, and a topic is a multinomial distribution over words. Topic Modeler works by learning and inferring the topic distribution of movies from their critic reviews. We employ PLDA that allows us to use the well-structured attributes (i.e. genre, director, actor, etc.) of movies to supervise the topic learning process, which contributes to higher quality and more interpretable learned topics. The attribute we choose is genre since genre has been proven to be a good indicator of users’ taste (Manzato 2012). PLDA regards genre as the high level category of movies, and learns the specified number of latent topics under each genre. It also uncovers the global shared background topics that may not belong to a specific genre. Table 1 shows some example topics automatically learned from the critic reviews. Each topic is represented as a set of most common words in this topic. As we can see, two global shared topics can be interpreted as “family” and “life” respectively, which mean that movies in different genres may all talk about family and life. The global shared topics are important since they capture the broad patterns across the whole corpus of critic reviews.

Specifically, we use $G$ to denote a set of genres and $G_i$ ($1 \leq i \leq |G|$) indicates the $i$ th genre. For each genre $G_i$, we assign some number of topics $T_{\alpha_i}$ to it, where each topic $T_{\alpha_i,j}$ ($1 \leq j \leq |T_{\alpha_i}|$) is a representation of a
multinomial distribution over all words in the vocabulary of the critic reviews. The number of topics for each genre can be different, which allows us to assign more topics to those genres having higher proportion of movies. In order to explore the global shared topics beyond the genres, a special label is used which can be interpreted as the “global” genre that is shared by all the movies, and some number of latent topics \( T_{\text{global}} \) are also assigned to it. PLDA is a generative model assuming that each word \( w \) in the critic reviews of a movie \( m \) belonging to a set of genres \( \Lambda_m \) are generated as follows: first, a genre \( g \) in \( \Lambda_m \) is drawn from a multinomial distribution of size \( |\Lambda_m| \), then a topic \( t \) in \( T_s \) is drawn from a multinomial distribution of size \( |T_s| \), and the word \( w \) is drawn from a multinomial distribution over the whole vocabulary in this topic. Intuitively, the probability that a word in the critic reviews of a movie is picked is in proportion to the aggregation of the following probabilities: (1) how likely this movie belongs to the genre \( g \); (2) how likely genre \( g \) belongs to the topic \( t \); and (3) how likely topic \( t \) has this word. The details of the algorithm for learning and inferring the model parameters can be found in (Asuncion et al. 2009).

We can use the critic reviews from a subset of the existing movies to build the topic model by learning the topic distribution. When a new movie is added, its critic reviews can be used to infer its topic distribution based on the learned topic model. The output of the Topic Modeler is matrix \( P \) representing the topic distribution of the movies. Each column of \( P \) is a vector \( \tilde{P}_m^T \) that represents the multinomial distribution over all topics for a movie \( m \), and each element \( P_{t,m} \) in this vector is the probability that movie \( m \) belongs to topic \( t \). All elements in \( \tilde{P}_m^T \) sum up to 1.

<table>
<thead>
<tr>
<th>Table 1. Example of Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(Global)</strong></td>
</tr>
<tr>
<td>Topic 1</td>
</tr>
<tr>
<td>Topic 2</td>
</tr>
<tr>
<td><strong>Drama</strong></td>
</tr>
<tr>
<td>Topic 1</td>
</tr>
<tr>
<td>Topic 2</td>
</tr>
<tr>
<td><strong>Comedy</strong></td>
</tr>
<tr>
<td>Topic 1</td>
</tr>
<tr>
<td>Topic 2</td>
</tr>
</tbody>
</table>

**Profile Learner**

Profile Learner is a core component in the recommendation engine. With the topic distribution of movies, Profile Learner utilizes the user ratings to learn the user preference by computing to what extent a given user likes a particular topic. Specifically, in order to isolate the users’ topic preference from other factors, we divide a user rating given to a movie into 4 parts: basis rating (i.e. overall average), user bias (i.e. some users may tend to rate higher or lower than other users), movie bias (i.e. some movies may tend to receive higher or lower ratings than other movies), and user topic preference. The original rating matrix \( X \) is approximated by:

\[
X \approx S + B_u + B_j + UI \tag{1}
\]

where each element \( X_{ij} \) in the matrix \( X \) is the rating given by user \( i \) to movie \( j \), all elements in \( S \) are equal to the global average rating \( \mu \), all elements in the \( i \) th row of matrix \( B_u \) have the same value that is equal to the user rating bias \( u_{bias_i} \), and all elements in the \( j \) th column of matrix \( B_j \) have the same value that is
equal to the movie rating bias $mbias_i$, $U \in \mathbb{R}^{m \times n}$ and $I \in \mathbb{R}^{n \times k}$, where $m$ is the total number of users, $n$ is the total number of movies, and $k$ is the total number of topics. $U_{ij}$ indicates the extent to which user $i$ prefers topic $t$, and $I_{t,j}$ indicates the extent to which movie $j$ belongs to topic $t$.

By adapting NMF, Profile Learner decomposes the original user-rating matrix into two matrices $U$ and $I$ to represent users’ topic preference and movies’ topic belonging respectively. Since the elements in the column of $P$ sum up to 1, we make a column-wise normalization for movie matrix $I$, and denote it as $\tilde{I}$. The decomposed matrices should satisfy two criteria: (a) the product of the user matrix $U$ and movie matrix $I$ should approximate the original matrix after adding the basis rating and rating bias; (b) the normalized movie matrix $\tilde{I}$ should approximate the topic distribution matrix $P$. The movie matrix $I$ acts a bridge between two types of data, i.e. user ratings and movie critic reviews, by satisfying the above-mentioned criteria. The first criterion can be satisfied by solving the least square error problem, and the second criterion can be satisfied by minimizing the Kullaback-Leibler (KL) divergence (Kullback 1987) between the normalized movie matrix $\tilde{I}$ and the topic distribution matrix $P$. The two criteria can be satisfied by solving the objective below:

$$\begin{align*}
\min_{b_i, b_j, U, I} & f(B_u, B_j, U, I) = \\
& \sum_{i=1}^{m}\sum_{j=1}^{n}\left( (X_{i,j} - S_{i,j} - B_{u,i} - B_{j,j} - (UI)_{i,j})^2 + \lambda_1[B_{u,i}^2 + B_{j,j}^2 + \sum_{t=1}^{k}(U_{i,t}^2 + I_{t,j}^2)] \right) + \lambda_2 D_{KL}(P || \tilde{I}).
\end{align*}$$

(2)

subject to $U_{i,j} > 0, I_{t,j} > 0, \forall i, j, t.$

where $\lambda_1$ indicates the extent of penalizing the magnitudes of the parameters to avoid over fitting, and $\lambda_2$ indicates the weight given to the topic distribution of critic reviews.

The values of the elements in $U$ and $I$ are initiated by assigning a random value $s$ ($0 < s < 0.1$) that follows Gaussian distribution. The movie rating bias and user rating bias are initiated as the average deviation from the global average rating $\mu$ with regularization parameters $\lambda_0$ and $\lambda_4$ as follows:

$$mbias_i = \frac{\sum_{j=1}^{n}(X_{i,j} - \mu), \text{ if } X_{i,j} \neq 0}{\lambda_0 + \# \text{non-zero elements in } X_i^T}.$$  

(3)

$$ubias_i = \frac{\sum_{j=1}^{n}(X_{i,j} - \mu - mbias_i), \text{ if } X_{i,j} \neq 0}{\lambda_4 + \# \text{non-zero elements in } X_i^T}.$$  

(4)

To satisfy the non-negative constrain, we employ a project gradient method to update the parameters. The detail of the algorithm can be found in (Lin 2007).

**Recommender**

Recommender is the only component running online, while the other components can run offline. The objective of Recommender is to match the user preference to the movie features in terms of topics and to generate movie recommendations for the users efficiently. For the existing movies, we can predict the users’ ratings by using the approximation: $S + B_u + B_j + UI$. For the new movies, we don’t have the item matrix $I$ or the movie rating bias $B_I$ since no user rating is available for them so that we cannot predict the users’ real ratings, but we can still estimate the users’ preference by using the topic matrix $P$ instead of $I$. However, the scale of the predicted ratings given by $UP$ for new movies is different from that given by $S + B_u + B_j + UI$ for existing movies. In order to unify the scale of predicted ratings and to make the existing and new items comparable, we predict another rating for each existing movie using the product of the user matrix $U$ and the normalized item matrix $\tilde{I}$, and recommendations are generated by selecting the items having highest predicted ratings given by $UP$ (for new movies) and $UI$ (for existing movies).
For example, if we want to predict the rating of movie $j$ for user $i$, user $i$’s topic preference is the $i$th row of user matrix $U$ shown as $\bar{U}_i$ in Table 2, $\bar{P}_j^r$, $\bar{I}_j^r$ and $\bar{I}_j^i$ in Table 2 are the $j$th columns in movie topic matrix $P$, movie matrix $I$ and normalized movie matrix $\bar{I}_j$ respectively, assuming that the overall average rating $\mu=2.5$, user $i$ tends to rate 0.5 higher than other users, i.e. $ubias_i=0.5$ and movie $j$ tends receive ratings that are 0.2 lower than other movies, i.e. $mbias_j=-0.2$, then the user’s rating on this movie is predicted by: $\mu + ubias_i + mbias_j + \bar{U}_i . \bar{I}_j^r = 2.95$. In order to make movie $j$ comparable with the new movies, another predicted rating is given by: $\bar{U}_i . \bar{I}_j^i = 0.313$. Supposed that movie $j$ is a new movie and we don’t have the matrix $I$ and $mbias_j$, the predicted rating is given by $\bar{U}_i . \bar{P}_j^r = 0.318$. The Recommender predicts the ratings for all the movies that the target user has not known, and then recommends the movies by providing the ranking list according to the predicted ratings.

<table>
<thead>
<tr>
<th>Table 2. Example of Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Table" /></td>
</tr>
</tbody>
</table>

The user ratings are stored in a $m \times n$ matrix ($m$ is the user number and $n$ is the movie number). If we apply the traditional Collaborative Filtering algorithm (Sarwar et al. 2001) to predict the rating of a given movie for a user using the original matrix, the time complexity is $O(m \times n \times k)$ where $k$ is the pre-defined neighborhood number. Our method decomposes the original matrix into two sub-matrices. The rating prediction is given by the dot product of two vectors whose length equals to the total number of topics $|T|$ ($|T| < m \times n$), and the time complexity becomes $O(|T|)$, therefore the rating prediction process in this online component can be efficient.

**Experiment and Results**

In this section, we are going to describe the experiment and the results to show the effectiveness of our proposed method. We will start from the evaluation metrics, and then will proceed to introduce the data set used and the configuration of the experiment, followed by the results of the experiment that compares the prediction error and ranking accuracy with state-of-the-art methods in recommending existing items, and evaluates the ranking accuracy in new item recommendation.

**Evaluation Metrics**

The accuracy of rating prediction is the most discussed property in recommendation research. Most research in recommender systems relies on a basic assumption that recommender system providing “accurate predictions” would be preferred by users (Shani and Gunawardana 2011), and seeks for algorithms that provide more accurate rating predictions. In line with this, we choose a commonly used metric in recommendation research, i.e. Mean Absolute Error (MAE) (Herlocker et al. 2004), to evaluate the accuracy of rating prediction. MAE is defined as:
where \( r_{u,i} \) is the rating given by user \( u \) to item \( i \) in the testing dataset, \( \hat{r}_{u,i} \) is the predicted rating and \( |\text{TestingSet}| \) is the size of testing dataset.

Although accurate prediction is crucial, in most cases, the recommendations are presented to the users as a list of items, and the order of items in the list is also important. Some research found that accurate prediction does not guarantee the correct order of the recommendations (McNee et al. 2006). A good RS should not only provide accurate rating predictions, but also should rank the recommended items correctly. In our experiment, we use the NDCG@k (Järvelin and Kekäläinen 2002) that is also a commonly used metric in recommendation to measure the ranking accuracy. NDCG@k is defined as:

\[
\text{NDCG@k} = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in \text{Top}k} \frac{2^{r_{u,i} - \text{cutOff} - 1}}{\log(1 + p)}.
\]

where \( U \) is the set of users, \( Z_n \) is a normalization factor to guarantee that for perfect ranking the NDCG value is 1, \( p \) is the position of recommended item in the list, and \( r_{u,p} \) is the rating given by the user \( u \) to the item at position \( p \).

**Experiment Setup**

In order to compare our method with other methods in the experiment, we use MovieLens dataset that is publicly available and is widely used in other research. The dataset consists of 100,000 ratings given by 943 users to 1682 movies. The user ratings are on a scale of 1 to 5, with 1 being bad and 5 being excellent.

The percentage of missing ratings in the dataset (aka. sparsity level) is \((1 - \frac{100000}{1682 \times 943}) \times 100\% = 93.69\%\).

The dataset also provides some movie information such as title, release year and genre. We use the title and release year to get the URLs of critic reviews via the API provided by Rotten Tomatoes, and crawl the corresponding critic reviews from external websites. 98.81% of the movies in the dataset have critic reviews in Rotten Tomatoes.

Before conducting the experiment, we need to assign the number of topics to each genre, and determine the values of some parameters in Profile Lerner. We vary the topic number to search the optimal number of topics that ensures most of the generated topics are interpretable. We assign 4 topics as the global shared latent topics, and for other genres, the number of topics is in proportion to the number of movies in this genre (i.e. one topic for 28 movies). For example, we assign 4 topics to the genre “children” that has 119 movies, and assign 2 topics to the genre “musical” that has 56 movies. There are 17 genres and the total number of topics is 68. We also vary the parameters to find the settings giving the best results. This occurs when \( \lambda_1 = 0.02 \), \( \lambda_2 = 0.5 \), \( \lambda_3 = 25 \) and \( \lambda_4 = 10 \). We use the same configuration in all the following experiments.

**Experimental Results**

In the field of RSs, many state-of-the-art approaches have been proposed in recent years. It is fair to compare our work with these methods, but the complexity and unclear description in the original publications of these methods make it difficult to re-implement all of them. A better way to do the comparison is to conduct the experiment in the same settings of these methods, and compare the results of our method with the reported results from the other methods. Although most methods report results in only one evaluation dimension, it is reasonable to compare with these results since we believe that in the evaluation dimension reported, these methods have the best results. In the following experiments, we denote our proposed method as **CRT** (Critic Review Topic) method.
Comparison on Prediction Errors

In this experiment, we compare the prediction errors of our CRT method with other state-of-the-art methods that have been reported to have good results in rating prediction. These methods for comparison are:

- CBS (Xue et al. 2005): This is a cluster-based smoothing method. It fills in the missing values by using other users' ratings in the same user cluster.
- WLR (Srebro and Jaakkola 2003): This method uses weighted low-rank approximation to fill in the missing values.
- CBT (Li et al. 2009): This method expands the codebook to reconstruct the rating matrix that is used to fill in the missing values.
- SVD++ (Koren 2008): This method reduces the dimension of the original matrix through Singular Value Decomposition. It also integrates the user feedback.
- TyCo (Cai et al. 2013): This method applies LDA to model keywords of movies and then constructs item typicality for further recommendation.

To make our result comparable, the same as (Li et al. 2009) and (Cai et al. 2013), we randomly select 500 users from the dataset, then use the first 100, 200 and 300 of them to form the training sets, named ML100U, ML200U and ML300U respectively. The last 200 users are used as testing set. For every user in the testing set, we keep 5, 10 and 15 ratings given by him in the training set, named as G5, G10 and G15 respectively. The training sets that have fewer users and fewer ratings from the test users are sparser. E.g. ML100U-G5 has the highest sparsity and ML300U-G15 has the lowest sparsity. Table 3 shows the comparison of our CRT method with the state-of-the-art methods on MAE. The results of other methods are reported in (Li et al. 2009) and (Cai et al. 2013). The results show that in most settings, our method has lower prediction errors than other methods. Excluding our method, the TyCo method has the best results among the others. In rating-dense settings, e.g. ML300U-G10 and ML300U-G15, the prediction errors of our CRT method are very close to those of TyCo, while in rating-sparse settings, e.g. all ML100U and all G5, our method outperforms the TyCo. The results are consistent with our findings in the previous experiment that our method has strength in sparse settings. Specifically, our CRT method reduces the prediction errors of TyCo using ML100U-G5, ML200U-G5 and ML300U-G5 by 5.06%, 5.78% and 4.91% respectively.
Comparison on Ranking Accuracy

In this experiment, we test the ranking accuracy of our CRT method, and compare the results with those reported by state-of-the-art methods that have been proven to have good performance in ranking the recommended items. The methods to be compared with are:

- ASSOC (Deshpande and Karypis 2004): This method uses the association among items to perform top N recommendation.
- FREQ (Sueiras et al. 2007): This method builds a model based on the hitting-frequency to predict the user preference.
- PMF (Salakhutdinov and Mnih 2008): This method employs Probabilistic Matrix Factorization to utilize the relationship among users, items and ratings.

Following the experiment in (Xin et al. 2011), we randomly choose 600 users to form the training set and the remaining 343 users are put in the testing set. For every user in the testing set, 5, 10 and 15 ratings from him are given in the training set, named as G5, G10 and G15 respectively. The fewer ratings are given to the training set, the sparser it is. The results of NDCG@1, NDCG@3 and NDCG@5 using different training and testing sets are shown in Table 4. The results of other methods are reported in (Xin et al. 2011). The results show that in all settings except G15-NDCG@5, our CRT method has better results, especially in the sparse settings like G5. The results are aligned with our previous findings that the advance of our CRT method is more salient in sparse settings. Our CRT method increases the NDCG@1, NDCG@3 and NDCG@5 of PMF using the sparsest training set (i.e. G5) by 11.81%, 12.91% and 9.31% respectively.

<table>
<thead>
<tr>
<th></th>
<th>G5</th>
<th></th>
<th>G10</th>
<th></th>
<th>G15</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NDCG @1</td>
<td>NDCG @3</td>
<td>NDCG @5</td>
<td>NDCG @1</td>
<td>NDCG @3</td>
<td>NDCG @5</td>
</tr>
<tr>
<td>ASSOC</td>
<td>0.529</td>
<td>0.542</td>
<td>0.560</td>
<td>0.597</td>
<td>0.593</td>
<td>0.595</td>
</tr>
<tr>
<td>FREQ</td>
<td>0.642</td>
<td>0.600</td>
<td>0.596</td>
<td>0.636</td>
<td>0.607</td>
<td>0.610</td>
</tr>
<tr>
<td>PMF</td>
<td>0.635</td>
<td>0.612</td>
<td>0.623</td>
<td>0.644</td>
<td>0.646</td>
<td>0.654</td>
</tr>
<tr>
<td>CRT</td>
<td>0.710</td>
<td>0.691</td>
<td>0.681</td>
<td>0.709</td>
<td>0.694</td>
<td>0.679</td>
</tr>
</tbody>
</table>

Comparison on New Item Recommendation

One of the key features of our method is that it supports new item recommendation, while all the above-mentioned state-of-the-art methods cannot work with new items. To illustrate the effectiveness of our CRT method in recommending new movies, we compare with another method TSCF (Spaeth and Desmarais 2013) that computes the text similarity between the item profiles (here we use the movie plot summaries in IMDB4), and then performs CF recommendation. In order to simulate the new movies, we randomly select 200,400 and 600 movies as new movies for testing, named as ML200M, ML400M and ML600M respectively, and the remaining movies are used as existing movies for training. For those movies in the testing set, none of their rating is given in the training set. Since the scale of the predicted ratings for new movies given by our method is different from that of the users’ real ratings, we do not compare the prediction errors here and only report the results of NDCG@k that are shown in Table 5. The results indicate that as the proportion of new movies increases, the ranking accuracy of both methods...

4 http://www.imdb.com
decreases, but our CRT method always performs better than the TSCF method. The results prove that our CRT method is effective in new item recommendation.

<table>
<thead>
<tr>
<th>Table 5. Comparison on New Item Recommendation (NDCG@k)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ML200M</strong></td>
</tr>
<tr>
<td>NDCG @1</td>
</tr>
<tr>
<td>TSCF</td>
</tr>
<tr>
<td>CRT</td>
</tr>
</tbody>
</table>

**Conclusion**

In this paper, we propose a novel content-based recommendation framework. A distinct feature of our method is that it incorporates the topics inferred from the external critic reviews of items to boost new item recommendation. We employ an advanced semi-supervised topic modeling approach, i.e. PLDA, which is able to uncover the global shared latent topics as well as the topics under each well-structured item attribute, to learn and infer the topic distribution of the critic reviews. We also adapt NMF to our context by redefining the error function to fully utilize the user ratings and topic distribution of critic reviews. The topics inferred from is critic reviews are better representations of the items since it covers more characteristics of the items and reflect more aspects of user tastes. By fully utilizing the user ratings and the inferred topics, our method alleviates the dependency on user ratings and enables high quality recommendation with new items. The adaption of NMF lowers the dimension of the original rating matrix, which contributes to high efficiency. The results of the experiment show that our proposed method is scalable and outperforms the state-of-the-art methods in terms of prediction accuracy and ranking accuracy, and the advance of our method is more salient in rating-sparse settings. Our method also generates high quality new item recommendation which is not supported by many state-of-the-art methods.

There are some limitations of our work. First, although critic reviews are available in abundance for new items and this work has proved that purely using critic reviews is effective in boosting new item recommendation, there do exist some items that do not have any critic review and cannot get recommended. Future work can increase the coverage of new items by incorporating more information in addition to critic reviews (e.g. the text description of items) into a unify framework. Second, off-line processing procedures (i.e. topic learning) are time consuming. In future work, parallel computation framework (e.g. MapReduce) can be adopted to accelerate the computation for large scale application. Third, we only use one kind of attribute (i.e. the genre of movie) to supervise the topic learning and inferring. Future work can try to use more attributes and to investigate how to integrate the topics under different attributes.

Although we focus on movie domain in this paper, our method may be generally applicable in any domain of consumer products where high quality critic reviews are available (e.g. books, digital products etc.). One possible extension to our work is to see whether our method can be applied in cross-domain recommendation.

**References**


