Semantic Context-aware Recommendation via Topic Models Leveraging Linked Open Data

Mehdi Allahyari and Krys Kochut

Computer Science Department
University of Georgia
Athens, GA
{mehdi,kochut}@cs.uga.edu

Abstract. Context aware recommendation systems are used to provide personalized recommendations by exploiting contextual situation. They take into account not only user preferences, but also additional relevant information (context). Statistical topic models such as Latent Dirichlet Allocation (LDA) have been extensively used for discovering latent semantic topics in text documents. In this paper, we propose a probabilistic topic model that incorporates user interests, item representation and context information in a single framework. In our approach, the contextual information is represented as a subset of the items feature space which is acquired from the knowledge available in the `Linked Open Data (LOD)`. We use `DBpedia`, a well-known knowledge base in `LOD`, to utilize the context information in recommendation. Our proposed recommendation framework computes the conditional probability of each item given the user preferences and the additional context. We use these probabilities as recommendation scores to find *top-n* items for recommendations. The performed experiments demonstrate the effectiveness of our proposed method and shows that leveraging semantic context from the `Linked Open Data` can improve the quality of the recommendations.

1 Introduction

Since its introduction, the amount of data published on the Web has grown dramatically. As a result of this information explosion, it has become very difficult for the users to find appropriate items relevant to their needs. Recommender Systems (RS) are widely used as some of the most essential techniques for *information filtering*. They help users in making decisions and finding what is relevant to them in a personalized manner. Recommender systems have also been successfully employed in the industry, for example, for product recommendations at Amazon and movie recommendations at Netflix. Recently, there has been a substantial amount of research on various recommendation techniques [9, 26, 35, 27, 3, 24, 33].

Although recommender systems are broadly used in multiple domains, they mostly do not consider the contextual situation in which the item is evaluated or used. Incorporating additional contextual information, such as time, location and other factors into the recommendation process can significantly increase the quality of the recommendations in many cases. For example, taking temporal context into consideration, a movie recommender system is able to provide movie recommendations for weekends that can be entirely different from the recommendations for the week days. In order to address this issue, *context-aware recommender systems* (CARS) have been introduced in recent years. CARS have shown to improve the accuracy and provide more relevant recommendations [6, 7, 25, 8].

One of the most commonly used definitions of context, proposed by Abowd et al. [1], states that it is *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*. Apart from this general definition of context, some more specific definitions were proposed in recent years. For example, Cantador and Castells [15] focused on semantic contextualization and represent context as *"the background topics under which activities of a user occur within a given unit of time"*.

Incorporation of context into the recommender systems can be done in a variety of ways. In general, the representation and integration of context rely on the available contextual information and the way context is defined. Typically, there are two approaches to acquire the context. In the first, users specify the context *explicitly* each time they interact with the system [17, 5]. For example, in a music recommender system, users can describe their current interests in a particular genre of music by giving that information as a query to the system. The recommender system then, recommends songs that best match the user's needs considering the previously established user preferences as well as the queried context. Although, explicit context assumption simplifies the system, it does not hold for many applications where the context is hidden and should be inferred. In the second approach, contextual information is *implicitly* inferred from the user's behavior [16, 20]. As an example, a restaurant recommender may produce recommendations considering implicitly variables such as time of the day and user's location along with past user's interests. In this paper, we consider a setting where context is not pre-specified, but rather is learned from the knowledge existing in the knowledge bases.

Within the Semantic Web, numerous data sources have been published as ontologies. Many of them are interconnected which have created a huge decentralized knowledge base commonly known as `Linked Open Data` (`LOD`) [10]. For example, DBpedia [11] (as part of LOD) is a publicly available knowledge base extracted from Wikipedia in the form of an ontology of concepts and relationships, making this vast amount of information programmatically accessible on the Web. This freely available knowledge has been used in several works to improve the quality of the recommender systems [30, 18, 31, 34, 14, 29, 4].

We propose a single probabilistic topic model that integrates user preferences, item descriptions and contextual information based on sound principles. We represent the context information as a subset of features representing the items.

In our model, the context is acquired from the semantic descriptions of the items obtained through DBpedia. We should point out that there exist several other knowledge bases such as YAGO [22], and Freebase [13] that could be exploited as the prior knowledge in our work. For this research, we selected DBpedia as arguably more frequently used for Semantic Web tasks, but our approach could be used with other knowledge bases, as well.

In our semantic context-aware recommendation system (SCRM) each user profile is represented as a multinomial distribution over a set of latent topics, while topics are distributions over items and item features. The main difference between our work with all prior works is that we propose a probabilistic model that both *infers* the semantic context and *models* this context in a systematic way. For a given user's profile u and context c, we compute the recommendation score for each item v as $p(v|c, u)$, rank the items based on these scores and select the *top-n* recommendations for the user.

The paper is organized as follows. In section 2, we discuss the prior work on context-aware recommendation systems. In section 3, we present a brief overview of Latent Dirichlet Allocation (LDA), the state-of-art probabilistic topic modeling technique. We formally define our semantic context-aware recommendation model in section 4. In section 5, we demonstrate the effectiveness of our method on a real-world dataset. Finally, we present our conclusions and future work in section 6.

2 Related Work

Several approaches have been recently proposed that make use of contextual information in recommender systems. [7] proposes a context-aware matrix factorization method for rating prediction. The system proposed in [6] uses contextual factors such as "temperature" or "weather" to recommend places of interest. [28] introduces a context-aware system for recommending playlists to the users according to their moods. There is also prior works [38, 39, 37, 36] that propose spatiotemporal recommendation models exploiting location and time as context for recommendations. Our method is different with these works, since in our system the context is not pre-specified. Ma et al. [27] propose matrix factorization methods that exploit social information to improve the prediction accuracy of recommender systems. The main distinction of our system is that we do not incorporate social-context.

Extracting contextual information from unstructured text is relatively recent and has not been widely addressed. Aciar [2] presents a classification method to identify review sentences containing contextual information. Our work is different from this work as they do not incorporate the retrieved information in the recommendation system. [21] proposes a context-aware system for hotel recommendation that obtains contextual information by mining user reviews and combining it with user rating history to compute a utility function over a set of items. They represent the context as a distribution function over the set of "trip types" which are pre-determined. Then, using Labeled-LDA topic model as a multi-class supervised classifier, they find context distributions over trip types. Eventually, they define a context score for each context and combine these scores with a item-based kNN recommender system for recommendation. Our method is different from this approach in two ways. First, unlike [21], we learn the contextual information through DBpedia knowledge about items. Second, we propose a single topic model that combines the user, items and the context in a systematic manner. Hariri et al. [20] introduce a context-aware recommender system that is more similar to our system. However, their item features are known and available (e.g., item tags) whereas we model and learn them via existing knowledge in DBpedia.

3 Latent Dirichlet Allocation (LDA)

Probabilistic topic models are a set of algorithms that are used to uncover the hidden thematic structure from a collection of documents. The main idea of topic modeling is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words. The two main topic models are Probabilistic Latent Semantic Analysis (pLSA) [23] and Latent Dirichlet Allocation (LDA) [12]. Hofmann (1999) introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents. Blei et al. [12] extended this model by introducing a Dirichlet prior on mixture weights of topics per documents, and called the model Latent Dirichlet Allocation (LDA). In this section we describe the LDA method.

The Latent Dirichlet Allocation (LDA) [12] is a generative probabilistic model for extracting thematic information (topics) of a collection of documents. LDA assumes that each document is made up of various topics, where each topic is a probability distribution over words.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ is the corpus and $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$ is the vocabulary of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j), \sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words given the document is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{1}$$

The graphical model of LDA is shown in Figure 1(a) and the generative process for the corpus $\mathcal{D}$ is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \mathrm{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, \mathcal{D}\}$,
   (a) Sample a topic distribution $\theta_d \sim \mathrm{Dir}(\alpha)$

(b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
   i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
   ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

The joint distribution of the model (hidden and observed variables) is:

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{|\mathcal{D}|} P(\theta_d|\alpha) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right)$$

4 Semantic Context-aware Recommendation Model

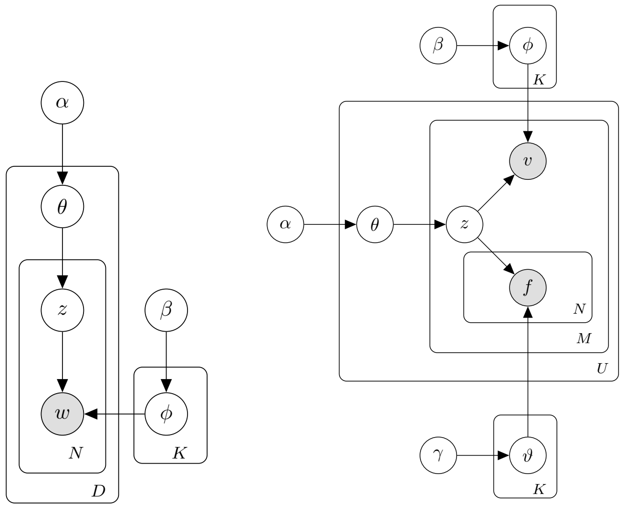
In this section, we formally describe our model. We propose a probabilistic topic model, which integrates the users, items and the contextual information in a unified framework. The underlying idea is that ontological knowledge existing in the LOD forms a *semantic context* which can be incorporated with user preferences to improve recommendations in collaborative filtering recommendation systems. Note, that our semantic context-aware recommendation model (SCRM) resembles the model introduced by Hariri et al. [20], with users being analogous to the corpus and user profiles being analogous to documents. Yet, our model differs from [20] in a way that we acquire the context from the information available in DBpedia. We describe items by using their related entities in the DBpedia ontology. Similarly for each item, the features are represented over its description. We run the standard LDA on the collection of items and then for each item, we extract a subset of its feature space (latent factors) and obtain the context. Details are explained in section 4.2.

The graphical representation of the SCRM is shown in Figure 1(b) and the generative process is defined as follows:

1. For each topic $k \in \{1, 2, \ldots, |K|\}$, draw an item distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each topic $k \in \{1, 2, \ldots, |K|\}$, draw a feature distribution $\vartheta_k \sim \text{Dir}(\gamma)$
3. For each user $u \in \{1, 2, \ldots, |U|\}$,
   (a) Draw a topic distribution $\theta_u \sim \text{Dir}(\alpha)$
   (b) For each of the M_u items v_i of user's profile u,
      i. Draw a topic $z_i \sim \text{Mult}(\theta_u)$
      ii. Draw an item $v_i \sim \text{Mult}(\phi_{z_i})$
      iii. For each of the N_{v_i} features f_i of item v_i, draw $f_i \sim \text{Mult}(\vartheta_{z_i})$

The generative process for the SCRM model corresponds to the following joint distribution of the hidden and observed variables:

$$P(\phi, \vartheta, \theta, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{f}) = \prod_{k=1}^{|K|} p(\phi_k|\beta) \prod_{k=1}^{|K|} p(\vartheta_k|\gamma) \prod_{u=1}^{|U|} p(\theta_u|\alpha) \prod_{i=1}^{|M|} \left(p(z_i|\theta_u) p(v_i|z_i, \phi) \prod_{j=1}^{|N|} p(f_j|z_i, \vartheta) \right) \quad (2)$$



(a) LDA model (b) SCRM model

Fig. 1. Graphical representation of different models

4.1 Inference and Estimation

Since the posterior inference of SCRM is intractable, we need to find an algorithm for estimating this posterior inference. A variety of algorithms have been used to estimate the parameters of topic models, such as variational EM [12] and Gibbs sampling [19]. In our SCRM topic model presented in this paper, we use the collapsed Gibbs sampling procedure. Collapsed Gibbs sampling [19] is a Markov Chain Monte Carlo (MCMC) algorithm, which constructs a Markov chain over the latent variables in the model and converges to the posterior distribution after a number of iterations. In our case, we aim to construct a Markov chain that converges to the posterior distribution over $\boldsymbol{z}$ conditioned on the observed items $\boldsymbol{v}$, features $\boldsymbol{f}$ and hyperparameters α, β and γ.

$$P(\boldsymbol{z}|\boldsymbol{v},\boldsymbol{f},\alpha,\beta,\gamma) = \frac{P(\boldsymbol{z},\boldsymbol{v},\boldsymbol{f}|\alpha,\beta,\gamma)}{P(\boldsymbol{v},\boldsymbol{f}|\alpha,\beta,\gamma)} \propto P(\boldsymbol{z},\boldsymbol{v},\boldsymbol{f}|\alpha,\beta,\gamma)$$
$$\propto P(\boldsymbol{z})P(\boldsymbol{v}|\boldsymbol{z})\prod_{i=1}^{N} P(f_i|\boldsymbol{z})$$

Let $c = \{f_1, f_2, \dots, f_{|N|}\}$. Subsequently, the update equation for the hidden variable can be derived as:

$$P(z_i = k|v_i = v, c_i = c, \boldsymbol{z}_{-i}, \boldsymbol{v}_{-i}, \boldsymbol{c}_{-i}, \alpha, \beta, \gamma) \propto$$
$$\frac{n_{k,-i}^{(u)} + \alpha}{\sum_{k'}(n_{k',-i}^{(u)} + \alpha)} \times \frac{n_{v,-i}^{(k)} + \beta}{\sum_{v'}(n_{v',-i}^{(k)} + \beta)} \times \prod_{f \in c} \frac{n_{f,-i}^{(k)} + \gamma}{\sum_{f'}(n_{f',-i}^{(k)} + \gamma)} \quad (3)$$

After Gibbs sampling, we can easily estimate the topic-item distributions ϕ, topic-feature distributions ϑ, and user-topic distributions θ by:

$$\phi_{kv} = \frac{n_v^{(k)} + \beta}{\sum_{v'} (n_{v'}^{(k)} + \beta)} \quad \vartheta_{kf} = \frac{n_{f,-i}^{(k)} + \gamma}{\sum_{f'} (n_{f',-i}^{(k)} + \gamma)} \quad \theta_{uk} = \frac{n_k^{(u)} + \alpha}{\sum_{k'} (n_{k'}^{(u)} + \alpha)} \tag{4}$$

where ϕ_{kv} is the probability of an item given a topic, ϑ_{kf} is the probability of a feature given a topic and θ_{uk} is the probability of a topic given a user.

4.2 Context Extraction from DBpedia

In this section, we describe how to extract the contextual information about items from the DBpedia ontology. The intuition is that items have a set of hidden features (e.g. topics) that can be learned or uncovered. We assume that knowledge bases such as DBpedia can be utilized to discover the features. Subsequently, the contextual information can be extracted as a subset of these features, and be used in context-aware recommendation systems to improve personalized recommendations. Since in this paper the application of the recommendation is for the movie domain, we regard movies as items. However, it should be noted that our overall approach is domain independent and applicable to other knowledge domains.

DBpedia is a publicly available knowledge base belonging to the LOD cloud which covers a diverse range of domains. In DBpedia, the knowledge about an entity, for example, a movie, a person, or a song is represented by linking different entities (vertices) to each other via semantic relationships (edges). Given an entity representing an item, its *description* includes the knowledge associated with it, i.e., related entities. In order to create an item description, in addition to DBpedia's *object properties*, we exploit other properties, including the `dcterms:subject` property, since these properties convey rich ontological knowledge about the entities. On the other hand, we exclude the properties `dbpedia-owl:thumbnail` and `dbpedia-owl:wikiPageExternalLink` since they do not give useful semantic information about the entities. Figure 2 shows a snippet of the description for the entity *"back_to_the_future"*.

For each movie, we extract all the related entities such as movies, actors, genres, directors, etc. from DBpedia and create a bag of entities. In our approach, each item corresponds to a document where entity labels represent the words of the document. Moreover, we consider the set of item features as latent variable $T = \{f_1, f_2, \ldots, f_{|T|}\}$ that needs to be discovered. We assume each item is a distribution over the set of features, while each feature is a distribution over the words. We run the LDA model and extract the set of features T for the collection of items. The probability of each feature f_i under item v, $p(f_i|v)$, indicates the significance of the f_i for item v.

We assume that the contextual information is represented as a subset of the items feature space. Therefore, for each item v, $c_v = \{f_{1v}, f_{2v}, \ldots, f_{|N|v}\}$ where $|N| \leq |T|$, i.e. the context c_v consists of the top-N features having the highest marginal probability under v.

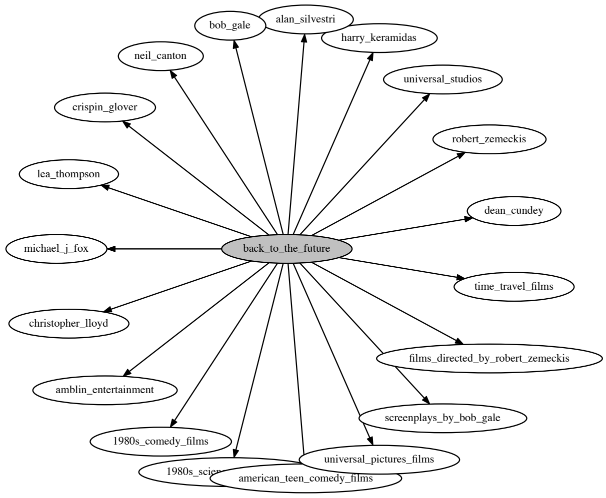


Fig. 2. Snippet of the description for the entity *"back_to_the_future"*

4.3 Semantic Context-aware Ranking Score

In this section, we detail the computation of the ranking score used in our SCRM model. Let $c = \{f_1, f_2, \ldots, f_{|N|}\}$ be a given context for user u with n items. We define a scoring function to rank the items and provide the personalized recommendations for user u considering the given context.

Given our model, for an item v, context c and user u, $p(v|c,u)$ is computed and used as the *ranking score*. Thus, according to Bayes rule:

$$p(v|c,u) \propto p(v|u) \cdot p(c|v,u) \tag{5}$$

where $p(v|u)$ and $p(c|v,u)$ are computed as follows:

$$p(v|u) = \sum_{k=1}^{K} p(v|z_k) \cdot p(z_k|u) = \sum_{k=1}^{K} \phi_{kv} \cdot \theta_{uk} \tag{6}$$

$$\begin{aligned} p(c|v,u) &= \prod_{f \in c} p(f|v,u) = \prod_{f \in c} \sum_{k=1}^{K} p(f|z_k) \cdot p(z_k|v,u) \\ &= \prod_{f \in c} \sum_{k=1}^{K} \vartheta_{kf} \cdot p(z_k|v,u) \end{aligned} \tag{7}$$

$$p(z_k|v,u) = \frac{p(z_k,v,u)}{p(v,u)} = \frac{p(v|z_k)\cdot p(z_k|u)}{\sum_{i=1}^{K} p(v|z_i)\cdot p(z_i|u)}$$
$$= \frac{\phi_{kv}\cdot\theta_{uk}}{\sum_{i=1}^{K}\phi_{kv}\cdot\theta_{uk}} \tag{8}$$

Combining equations 6, 7 and 8, equation 5 can ber simplified as:

$$p(v|c,u) \propto \left(\sum_{k=1}^{K}\phi_{kv}\cdot\theta_{uk}\right)\cdot\prod_{f\in c}\sum_{k=1}^{K}\frac{\vartheta_{kf}\cdot\phi_{kv}\cdot\theta_{uk}}{\sum_{i=1}^{K}\phi_{kv}\cdot\theta_{uk}} \tag{9}$$

5 Experiments and Results

In this section, we evaluate our method and compare it with several baselines. The evaluation has been carried out on a real-world data set `MovieLens` from the movie domain with ratings in $\{1,2,3,4,5\}$.

5.1 Dataset

We performed the evaluation on `MovieLens` 1M dataset[1], one of the most commonly used datasets for movie recommender systems. This dataset contains $1,000,209$ ratings of roughly $3,900$ movies made by $6,040$ users. Since our method is based on semantic context-aware recommendation, in order to use this dataset, we need to link each movie in `MovieLens` to the corresponding entities in DBpedia. However, Noia et al. [18] have carried out the mappings and created a mapping file which is publicly available[2].

We represented the user profile based on a binary rating such as *like/dislike*. Nonetheless, in `MovieLens` user u rates a movie based on a five-value scale: $r(u,v) \in \{1,2,3,4,5\}$ where a rate 1 indicates a *terrible* movie whereas a rate 5 implies an *excellent* movie. In order to map the five-scale ratings to a binary one, we considered the ratings above 3 as *like* and the others as *dislike*. Therefore, we model the user profile as:

$$profile(u) = \{v_i | r(u,v_i) > 3\} \tag{10}$$

5.2 Experimental Setup

For evaluation, we performed a 5-fold cross validation. In our setting, 80% of the items in each user's profile used for training the model, while the remaining 20% were put aside for testing. We also used the method explained in section 4.2 to obtain the contextual information for the movies in the dataset. For each user u,

1 `http://grouplens.org/datasets/movielens/1m/`
2 `http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets`

held-out item v and context $c_v = \{f_{1v}, f_{2v}, \ldots, f_{|N|v}\}$ (consisting of $|N|$ features f_i), we compute a recommendation score from each of the compared methods. As described in section 4.3, our SCRM model computes the ranking score for each item v_i as $p(v_i|c_{v_i}, u)$ and sorts the items in descending order based on these scores. We compared our proposed SCRM model with the following baselines:

1. **User-based kNN:** This approach is one of the most commonly used collaborative filtering methods where items are recommended to a user based on similar user profiles using the k-nearest neighbor approach.

2. **Item-based kNN:** This approach applies the same idea as **User-based kNN**, but exploits the similarity between the items instead of the users.

3. **BPRMF:** This is a matrix factorization method that is optimized based on Bayesian Personalized Ranking (BPR) criterion for personalized rankings of items [32].

For our SCRM model, we set the number of topics $K = 10$, and assumed the symmetric Dirichlet prior and set $\alpha = 50/K$, $\beta = 0.01$ and $\gamma = 0.01$, respectively. We ran the Gibbs sampling algorithm for 1000 iterations and computed the posterior inference after the last sampling iteration. For extracting the set of features from the items, we implemented the LDA model with the Mallet toolkit[3]. We set the number of features to $|T| = 10$, and all other settings the same as in the SCRM model. We also restricted the size of contextual features to $|N| = 1$, meaning for each item v, the context c_v consists of the top-1 feature having the highest marginal probability under v. For both user-based kNN and item-based kNN, we set the number of neighbors to $k = 10$.

5.3 Experimental Results

In this section, in order to show the the effectiveness of our proposed approach, we compare different algorithms in terms of their capability to provide personalized recommendation based on the given context. Thus, we measure how well they discover the movies in the test data. Since top recommendations are more important to the user, for each held-out movie, we find its rank in the overall recommendation list. We run a cross validation and evaluate the results by measuring the ***Hit Ratio*** metric. The hit ratio computes the probability that a removed movie is recommended as part of the *top-n* recommendations. If a removed movie is part of the *top-n* recommendations, we consider it a *hit.* We represent the *top-n* recommendations for a given user u as $R_n(u)$. For the held-out movie m_u, we define an indicator function:

$$\mathbb{1}_n(u, m_u) = \begin{cases} 1 & \text{if } m_u \in R_n(u) \\ 0 & \text{otherwise.} \end{cases}$$

[3] `http://mallet.cs.umass.edu/`

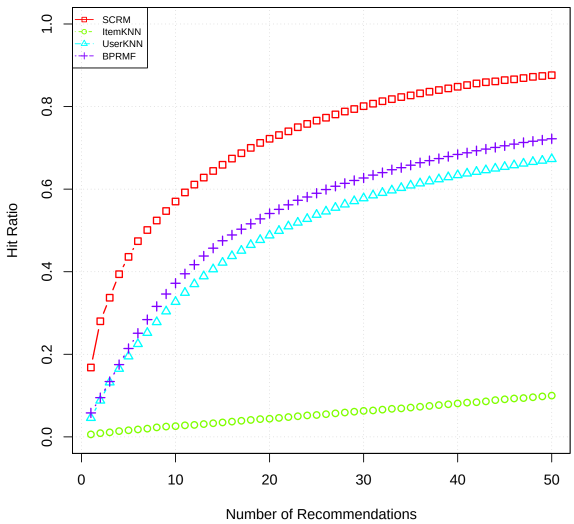


Fig. 3. Hit ratio for different number of recommendations for `MovieLens` dataset

For any given rank n, the hit ratio of the recommendation algorithm is defined as follows:

$$h(n) = \frac{\sum_{i=1}^{|U_{test}|} \mathbb{1}_n(u_i, m_{u_i})}{|U_{test}|} \tag{11}$$

where U_{test} is the set of users in the test set. Figure 3 illustrates the average hit ratio results (i.e. average of five-fold cross validation) of our approach as well as all other baselines for the first 50 recommendations. It shows that our SCRM model significantly outperforms the other methods at all ranks. Similarly, Table 1 presents the hit ratio measurement values of *top-n* where $n = \{1, 5, 10, 20, 30, 40, 50\}$, which shows that our approach achieves considerably better results. For example, if we consider the top-10 recommendations, the hit ratio of our approach is 20 times better than item-based kNN and 1.6 times higher than user-based kNN.

5.4 Context Analysis

As mentioned in section 4.2, context is represented as a subset of the features of items. In our setting, features are analogous to categories (topics) of the items and are learned through LDA model (i.e. items are grouped per item category). Thus, each item has a distribution over all the categories with different probabilities, and a category with a high probability under an item indicates that the item is classified under that category.

Table 1. Average hit ratio for *top-n* recommendations for various methods.

	n = 1	**n = 5**	**n = 10**	**n = 20**	**n = 30**	**n = 40**	**n = 50**
Item-based *k*NN	0.006	0.016	0.026	0.044	0.063	0.081	0.100
User-based *k*NN	0.046	0.195	0.327	0.488	0.578	0.634	0.673
BPRMF	0.058	0.214	0.372	0.541	0.627	0.684	0.722
SCRM	**0.168**	**0.436**	**0.570**	**0.722**	**0.801**	**0.848**	**0.876**

In our experiments, we set the size of the context $|N| = 1$, i.e. the topmost (top-1) feature having the highest marginal probability under a given item was considered as the context. In other words, for each item, we selected the topmost category. Figure 4 illustrates the hit ratio of our semantic context-aware model with varying sizes of the context. As can be seen, when we increase the size of the features in the context, the hit ratio drops. The reason is that growing the size of the context adds noise to the model, which impacts the accuracy of the recommendations. Baltrunas et al. [7] propose several context-aware matrix factorization methods and consider the interaction of context with items at different levels of granularity (i.e. contextual information with different sizes). They demonstrate that the model having one contextual parameter for each item category outperforms the other models with larger number of contextual factors. Additionally, they explain that the best number of contextual features with respect to the items, depends on the domain and the amount of data available. It should be noted that this is consistent with our assumptions that (1) the context is a subset of features for a given item and (2) setting the size of the context to $|N| = 1$ means that we select one contextual factor for each item category (i.e. the top feature as the context).

6 Conclusion and Future Work

In this paper, we proposed SCRM, a semantic context-aware recommendation system that integrates user profiles, item descriptions and contextual information in a unified probabilistic topic model. The system learns the context using the knowledge from the DBpedia ontology and utilizes this additional information to compute the ranking score for each item and provides personalized recommendations for users. We demonstrated that exploiting *semantic context* from ontologies can improve the recommendations, by conducting thorough experiments.

There are many interesting future directions of this work. We evaluated our approach on one data set in the domain of movies. As a future work, we plan to perform comprehensive evaluations on multiple datasets and extend our model to other domains such as music and scientific articles. Also, in this work, we did not use the social context, i.e. interactions between the users, in the recommender system. It would be interesting to investigate how to integrate social information network among users in the model. In the work presented here, we

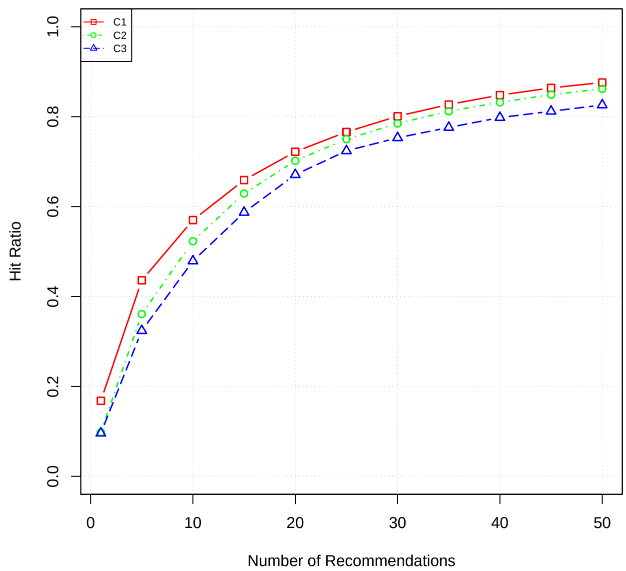


Fig. 4. Hit ratio with different sizes of the context. C_i shows that the size of the context is $|N| = i$

set the number of features to $T = |10|$. Hence, an interesting direction for future research is to explore approaches to determine the best number of features to consider. Furthermore, exploring a much richer set of semantic context-aware recommendation models that combine the contextual information, items and users would be a promising direction for future work.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Handheld and ubiquitous computing. pp. 304–307. Springer (1999)
2. Aciar, S.: Mining context information from consumers reviews. In: Proceedings of Workshop on Context-Aware Recommender System. ACM. vol. 201 (2010)
3. Agarwal, D., Chen, B.C.: flda: matrix factorization through latent dirichlet allocation. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 91–100. ACM (2010)
4. Anand, S.S., Kearney, P., Shapcott, M.: Generating semantically enriched user profiles for web personalization. ACM Transactions on Internet Technology (TOIT) 7(4), 22 (2007)
5. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices. Applied Artificial Intelligence 17(8-9), 687–714 (2003)

6. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing 16(5), 507–526 (2012)
7. Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation. In: Proceedings of the fifth ACM conference on Recommender systems. pp. 301–304. ACM (2011)
8. Baltrunas, L., Ricci, F.: Context-based splitting of item ratings in collaborative filtering. In: Proceedings of the third ACM conference on Recommender systems. pp. 245–248. ACM (2009)
9. Beutel, A., Murray, K., Faloutsos, C., Smola, A.J.: Cobafi: collaborative bayesian filtering. In: Proceedings of the 23rd international conference on World wide web. pp. 97–108. ACM (2014)
10. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. Semantic Services, Interoperability and Web Applications: Emerging Concepts pp. 205–227 (2009)
11. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web 7(3), 154–165 (2009)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research 3, 993–1022 (2003)
13. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. ACM (2008)
14. Cantador, I., Bellogín, A., Castells, P.: A multilayer ontology-based hybrid recommendation model. Ai Communications (2008)
15. Cantador, I., Castells, P.: Semantic contextualisation in a news recommender system. In: Workshop on Context-Aware Recommender Systems (CARS 2009) (2009)
16. Chen, G., Chen, L.: Recommendation based on contextual opinions. In: User Modeling, Adaptation, and Personalization, pp. 61–73. Springer (2014)
17. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 17–24. ACM (2000)
18. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 1–8. ACM (2012)
19. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National academy of Sciences of the United States of America 101(Suppl 1), 5228–5235 (2004)
20. Hariri, N., Mobasher, B., Burke, R.: Query-driven context aware recommendation. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 9–16. ACM (2013)
21. Hariri, N., Zheng, Y., Mobasher, B., Burke, R.: Context-aware recommendation based on review mining. General Co-Chairs p. 27 (2011)
22. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. pp. 3161–3165. AAAI Press (2013)

23. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 50–57. ACM (1999)
24. Jahrer, M., Töscher, A., Legenstein, R.: Combining predictions for accurate recommender systems. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 693–702. ACM (2010)
25. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 79–86. ACM (2010)
26. Leskovec, J., McAuley, J.: Hidden factors and hidden topics: Understanding rating dimensions with review text. Department of Computer Science, Stanford University (2013)
27. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the fourth ACM international conference on Web search and data mining. pp. 287–296. ACM (2011)
28. Meyers, O.C.: A mood-based music classification and exploration system. Ph.D. thesis, Massachusetts institute of technology (2007)
29. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. In: Web Mining: From Web to Semantic Web, pp. 57–76. Springer (2004)
30. Ostuni, V.C., Di Noia, T., Di Sciascio, E., Mirizzi, R.: Top-n recommendations from implicit feedback leveraging linked open data. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 85–92. ACM (2013)
31. Passant, A.: dbrecmusic recommendations using dbpedia. In: The Semantic Web–ISWC 2010, pp. 209–224. Springer (2010)
32. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 452–461. AUAI Press (2009)
33. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proceedings of the 25th international conference on Machine learning. pp. 880–887. ACM (2008)
34. Semeraro, G., Lops, P., Basile, P., de Gemmis, M.: Knowledge infusion into content-based recommender systems. In: Proceedings of the third ACM conference on Recommender systems. pp. 301–304. ACM (2009)
35. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 448–456. ACM (2011)
36. Yin, H., Cui, B., Chen, L., Hu, Z., Huang, Z.: A temporal context-aware model for user behavior modeling in social media systems. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 1543–1554. ACM (2014)
37. Yin, H., Cui, B., Chen, L., Hu, Z., Zhou, X.: Dynamic user modeling in social media systems. ACM Transactions on Information Systems (TOIS) 33(3), 10 (2015)
38. Yin, H., Cui, B., Sun, Y., Hu, Z., Chen, L.: Lcars: A spatial item recommender system. ACM Transactions on Information Systems (TOIS) 32(3), 11 (2014)
39. Yin, H., Zhou, X., Shao, Y., Wang, H., Sadiq, S.: Joint modeling of user check-in behaviors for point-of-interest recommendation. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1631–1640. ACM (2015)