Context-Aware Recommendation Based On Review Mining

Negar Hariri, Bamshad Mobasher, Robin Burke and Yong Zheng
DePaul University, College of Computing and Digital Media
243 S. Wabash Ave, Chicago, IL 60604, USA
{nhariri, mobasher, burke, yzheng8}@cs.depaul.edu

Abstract

Recommender systems are important building blocks in many of today's e-commerce applications including targeted advertising, personalized marketing and information retrieval. In recent years, the importance of contextual information has motivated many researchers to focus on designing systems that produce personalized recommendations in accordance with the available contextual information of users. Compared to the traditional systems that mainly utilize users' preference history, context-aware recommender systems provide more relevant results to users. We introduce a context-aware recommender system that obtains contextual information by mining user reviews and combining them with user rating history to compute a utility function over a set of items. An item utility is a measure that shows how much it is preferred according to user's current context. In our system, the context inference is modeled as a supervised topic-modeling problem in which a set of categories for a contextual attribute constitute the topic set. As an example application, we used our method to mine hidden contextual data from customers' reviews of hotels and use it to produce context-aware recommendations. Our evaluations suggest that our system can help produce better recommendations in comparison to a standard kNN recommender system.

1 Introduction

In recent years, recommender systems (RS) have been extensively used in various domains to recommend items of interest to users based on their profiles. A user's profile is a reflection of the user's previous selections and preferences that can be captured as rating scores given to different items in the system. Using preference data, different systems have been developed to produce personalized recommendations based on collaborative filtering, content-based or a hybrid approach.

Despite the broad usage of such recommender systems, failure to consider the users' current situations may result in considerable performance degradation in recommendations. For example, a customer who has once bought a toy for his friend's child may repeatedly receive suggestions to buy items related to kids as the recommendation algorithm decides based on the whole history in user's profile without prioritizing his current interests. To address this issue, the notion of context and context-aware recommender systems (CARS) has been introduced.

Contextual information can be explicit or implicit and can be inferred in different ways such as using GPS sensor data, clickstream analysis or monitoring user rating behavior. In this paper, we concentrate on deriving context from a textual description of a user's current state and the item features in which he/she is interested. This data can be in different forms such as tweets, blog posts, review texts or it can be given directly to the system as part of a query.

As an example application of our approach, we have used our method to mine hidden contextual data from customers' reviews of hotels. The reason behind the selection of this dataset is that users usually provide some contextual cues in their comments. For example, they may mention that they are with family or on a business trip, or they may express their opinions about the hotel services that are important to them such as having wireless internet, conference rooms, etc. In order to evaluate our method, we have used "trip Advisor" hotel reviews dataset where each review contains an overall rating, an optional review comment and also a "trip type" attribute that shows the types of trips user suggest for this hotel. For this attribute, the user can select a subset of five possible values: Family, Couples, Solo travel, Business, and Friends' getaway. The "trip type" attribute is not a feature of user or hotel (as different users may assign different values), it is rather related to the interaction and it is assumed to be an indication of context in our system.

Our approach in inferring context is based on using a classifier that is trained by the samples of descriptions and their corresponding contexts. Usually the trip type that a customer picks for a hotel is related to his review. Having this assumption, a set of review texts and their associated trip types are selected as the training set for the context classifier. After training, for a given description (as the user context) the classifier computes the probability of each the trip category. This probability distribution is used to infer context. Since we are dealing with a multi-class supervised classification problem, we chose to use Labeled-LDA [1] as our categorization method as based on our experiments it performs better in our dataset

in comparison to other similar methods.

We propose a method to use this inferred context to produce context-aware recommendations. While most of the existing approaches assume that a user's rating behavior depends on the current context and predict a rating function, we differentiate between the "rating" that a user gives to an item and the "utility" gains from choosing it. The inferred context is used to define a utility function for the items reflecting how much each item is preferred by a user given his current context. More specifically, the utility value depends on two factors: the predicted rating and the "*context score*" where context score represents the suitability of an item for a user in a given context. Rating can be predicted based on any conventional recommendation algorithms such as kNN.

Through the rest of this paper, we will first review some of the related work. Section 3 describes our proposed context-aware recommendation process. Finally, section 4 includes the evaluation of the proposed method and its comparison with traditional recommender.

2 Related Work

Several researchers have previously investigated the use of contextual information in various applications of recommender systems. Although there is no clear-cut definition of context, one of the most commonly used definitions was suggested by Abowd et al. [2] as follows: "Context 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." This is a general definition that limits the context only to the information that could be used to characterize the situation or the circumstance. Another similar definition by H.Lieberman et al. [3] is: "context can be considered to be everything that affects computation except the explicit input and output". In addition to these general definitions, a number of more specific definitions of context have been recently provided. For example, "Context can be described by a vector of context attributes, e.g. time, location or currently available network bandwidth in a mobile scenario". [4].

Capturing and representation of context in a system depends on the way context is defined in that system. Dourish et al. [5] presented two different views in modeling context: The representational view and the interactional view. In representational view, context is defined as *a form of information* that is *stable*, *delineable* and is *separate from activity*. Having this view, context can be defined and represented as a specific set of attributes of the environment within which the user's interaction with the system has taken place. For example, time and location can be considered as contextual attributes. In interactional view, it is assumed that *contextuality is a rational property* that holds between objects and activities rather than to be information (as in the case of representational view). Also, the contextual features are not definable and static but their *scope is defined dynamically*. Furthermore, rather than assuming that context defines the situation within which an activity occurs, it is assumed that *context arises from activity* and activity is induced by context. Therefore, even though context is not observable itself, the activity that arise from the context can be observed.

Adomavicius et al. [6] suggest three different architectures for context-aware recommender systems: In Contextual pre-filtering approach, the dataset is first filtered, the recommendations will be then provided based on the contextualized dataset. On the other hand, contextual post-filtering approach generates recommendations similar to traditional recommender systems. It will then filter and re-rank these recommendations to provide contextual recommendations. In contextual modeling, context is added to the problem as an additional dimension; meaning that in contrast to traditional recommender systems that estimate the rating function in two dimensional space of $User \times Item$, the context-aware recommender system is defined over the space of $User \times Item \times Context$. The representation of context and the way it should be captured and integrated into the recommendation algorithm depend on available contextual information as well as the definition of context in the system.

An interesting application of context-aware recommender systems is in mobile devices that are equipped with GPS or have internet access. In this case, different contextual information can be captured in real-time in order to be used in the recommendation process. For example PioApp Recommender [4] produces recommendations based on points of interest (such as restaurants, museum and train stations) in the neighborhood of the mobile user. Social camera introduced in [7] assists users in picking photo compositions given their current location and scene context. Many mobile travel applications such as [8–10] have also took advantage of context in order to make better suggestions. Numerous algorithms have also been suggested for music and movie recommendation (as well as many other domains). Micro-profiling introduced in [11], splits each single user profile into several possibly overlapping sub-profiles where each of them represent user's preference in a particular context. A context random walk algorithm was proposed in [12] to model the user's movie browsing behavior and then use it to make context-aware recommendations.

Some of the above mentioned approaches such as [8, 9] use a simple representational view of context where context is shown as a set of attributes (such as time, location, weather conditions) that is given to the system as input; while some other systems try to infer the contextual attributes from the user behavior. Instead of using a representational model, the context-aware recommender in [13] uses an interaction model. The proposed system was inspired by human memory model in psychology where the short term and long term memories are separately modeled. The short term memory contains the user preferences derived from his active interaction with the system while the long term memory stores the preference models related to his previous interactions with the system. They introduced three types of contextual cues including collaborative, semantic and behavioral cues in order to retrieve relevant preference models from the long term memory. The retrieved memory objects will then be combined with user's current preference model to generate and aggregate a final preference model that is used to produce recommendations.

In this paper we propose a method for mining contextual data from texual reviews. The importance of the hidden data in review comments has been the subject of many researches in the area of opinion mining and sentiment analysis. In opinion analysis, various Natural Language processing techniques and text analysis methods are applied to a set of review to extract attributes of the object that are referred to in the review text and to discover polarity (positive, negative or neutral) of the expressed opinions.

The problem of extracting contextual information from unstructured text is fairly new and has not been extensively addressed in prior researches. Aciar [14] introduces a method to identify review sentences which contain contextual information. In their approach, rule sets were created to classify sentences into contextual and preference categories where the preference category groups sentences including user's evaluation of the features. The approach presented in [14] does not discuss the use of the retrieved information in the recommendation process while we will provide a way of incorporating the contextual knowledge in producing the recommendations.

3 Context-Aware Recommendation Process

Our context-Aware recommender system (CARS), includes several components. The first component is the context miner that is responsible for determining a user's current context. Context is represented as a distribution function over the set of trip types and can be mined from a textual description of user's current situation and the features that are important to him. The main part of the context inference module consists of a multi-class supervised classifier. After training the classifier, context can be inferred for a given query. An example query is shown in Table 1. Based on the underlined words, it seems that the user is most probably looking for "couples" or "family" type trip rather than a "business" one.

I'm planning a romantic trip for my anniversary. I'm looking for an all inclusive resort near a beach. I expect the hotel room to be spacious, have a nice view over the sea and to be nicely decorated.

Table 1: A sample query

The second component of our system is the rating predictor that is a simple collaborative filtering recommender which predicts ratings of items. This component can be replaced with other types of rating prediction algorithms. The third component calculates the utility function based on user's current context and the predicted rating and presents a set of suggestions according to the order of the utility values.

3.1 Context Representation

Contextual recommender systems can either have an interactional or a representational view of the context. In this paper, we assume there are explicit labels representing context and the contextual information is obtained for each textual review by mapping it to this label set.

In our experiments, a dataset containing a set of hotel reviews from Trip Advisor website has been used. In this dataset, the "trip type" attribute assigned to a hotel review shows the types of trips that the user suggest for the hotel. The assigned attribute can be selected by the user from a set of five possible choices: Family, couples, Solo travel, business, and friends' getaway. We assume that this element is the representation of context in our system. A sample review from this dataset is depicted in Table 1. This sample shows the relationship between the trip type attribute and the review comment. For example "budget accommodation", "twin bedroom", "small" and "shared bathroom" can be more related to Friends getaway trip than to a business trip or a family travel.

Producing context aware recommendations requires mining the user's current context. If the user explicitly specifies his context, then it can be easily used in the recommendation algorithm. On the other hand, if he implies his context in a set of sentences describing his current state or his desired features for the hotel, then an inference method is required to determine the probability of each trip type. In this way, the context is shown as a distribution over the set of trip categories. In both cases, let $Context_u^i$ denote the context of user u when using item i. For example, if the reviewer u indicates the trip type for hotel i as business and solo travel, then the context represenation is $Context_u^i = \{P(\text{family}) = 0, P(\text{couples}) = 0, P(\text{solo travel}) = 0.5, P(\text{business}) = 0.5, P(\text{friends' get away}) = 0\}$.

The context inference problem just described is similar to a multi-labeled text classification problem in which documents can be a classified into one or more categories. The general solution is to provide a training set, build the model and use the model to categorize the new documents. If the trip type categories assigned to each review is assumed to be related to the review comment (and we will show they are related in our dataset), then we can use a set of review comments and their corresponding trip type values as our training set for training the classifier.

Trip type	Friends' getaway
Review Comment	This is an excellent option for budget accommodation in a hostel type establishment in a top class location, very close to central station and quick bus journey to circular quay. Stayed in twin bedroom which was very small but did the trick. If all you want is a clean bed in a clean room then this is grand. Shared bathroom and showering facilities were kept clean too.
Summary Quote	Excellent hostel accommodation in great location

Table 2: A sample review comment and the associated trip type

3.2 Inferring the Context

Different techniques have been used in text categorization such as probabilistic methods, regression modeling and SVM

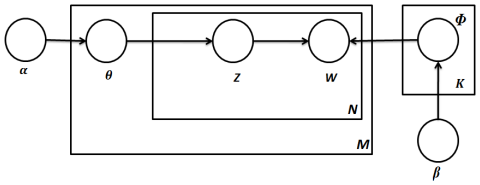

Figure 1: Graphical Representation of LDA [16]

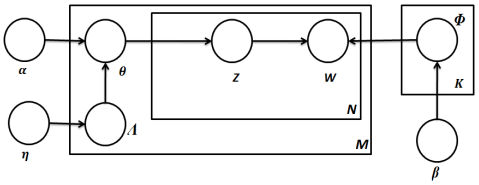

Figure 2: Graphical Representation of Labeled-LDA [1]

classification. In this article, we have used Labeled Latent Drichlet Allocation [1] (shown as L-LDA) for our dataset as it has shown to perform relatively well on our dataset. This method is a supervised classification algorithm for multi-labeled text corpora and is based on topic modeling.

Topic modeling and Labeled-LDA

Topic modeling deals with statistical modeling of documents in order to discover the latent topics behind them. Probabilistic latent semantic analysis (PLSA) [15] is one of the early approaches in this area which models a document as a probability distribution over the set of topics.

Later, the Latent Drichlet Allocation [16], known as LDA, was proposed as an extension of PLSA. LDA specifies a generative process for creating documents. The document generation is based on the idea that documents are mixture of topics, where a topic is a probability distribution over words. To generate a new document d, first the distribution over topics denoted by $\theta^{(d)}$ should be specified. For each word in the document a topic t is selected based on $\theta^{(d)}$. Let $\phi^{(t)}$ denote the multinomial distribution over words for topic t; According to this distribution a word is picked and is added to the document. It should be noted that this is similar to the general procedure followed by most of the existing topic models except that the statistical assumptions differs based on the model. The LDA model assumes that the topic mixture θ is a k-dimensional random variable as follows [16].

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1} \tag{1}$$

Where α is k-vector with elements $\alpha_i > 0$ and $\Gamma(x)$ is the gamma function. Figure 1 describes the graphical representation of LDA where the rectangles show replicates. The outer rectangle represents M documents while the inner rectangle illustrates the process of sampling words for a document of size N. In the LDA model, the document size follows a Poisson distribution. In corpora with a large vocabulary, it is likely that some of the words do not appear in the training examples. In order to cope with problem, a smoothing strategy is used by placing a Drichlet prior on ϕ with parameter β as shown in the figure.

In our problem, the user reviews are assumed to be documents where the topics behind these documents are the set of possible values for a trip type. As the topics are predefined, we need to adopt a supervised topic modeling approach. Some variations of LDA have been proposed to support supervised learning such as [1, 17, 18] among which we chose to use Labeled-LDA [1] as the other methods limit each document to be associated with only one topic while in our case, reviews can have multiple labels. Similar to LDA, in Labeled-LDA modeling, each word in the document is assigned a single topic. However, in order to incorporate supervision, the topic should belong to the label set of the document. In other words, there is a one-to-one relationship between the set of labels assigned to the documents and the topics and the topic mixture of each document is formed according to its label set. Figure 2 shows the graphical presentation of Labeled-LDA. Having k unique labels in all documents, the parameter Λ for each document is a k dimensional binary vector that shows the presents or absence of each topic in the document label set. For each document, Λ is generated using a Bernoulli coin toss with a prior probability vector η

As in [1], we used Gibbs sampling [19] for training. Let C^{WT} and C^{DT} represent two matrices which contain word-topic counts and document-topic counts respectively. Gibbs sampling begins with randomly assigning words to topics and filling the two matrices accordingly. Then iteratively updates them to finally converge to estimations of θ and ϕ. At each iteration, a word token is selected and its current topic is removed and C^{WT} and C^{DT} are updated by decrementing the corresponding entries to the removed topic assignment. Then, a new topic is sampled based on the topic assignments to all other words and the count matrices are incremented accordingly. After convergence, estimates of θ and ϕ can be obtained using equations 2 and 3 respectively.

$$\theta_j^{(d)} = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T\alpha} \tag{2}$$

$$\phi_i^{(j)} = \frac{C_{ij}^{WT} + \beta}{\sum_{k=1}^{W} C_{kj}^{WT} + W\beta} \tag{3}$$

3.3 Predicting item utility

As noted earlier, we make a distinction between predicting *rating* and *utility*. We assume that the utility of an item for a user may differ among different contexts, even though the user has rated the same item equally in those contexts. For instance, in the hotel review dataset it is possible that the rating given by a customer to hotel on a business trip does not change if he visits the same hotel one more time with his family while the utility of selecting that hotel changes from one trip type to the other. When he is on a business trip, business services of the hotel are more important while in a family trip

some other characteristics of the hotel (such as having a pool, distance to beach etc.) gain more priority.

We define *context score* as a measure of suitability of an item for a user in a given context. To calculate the context score for user u and item i, we need to predict the context that u would assign to i that is denoted by $predictedContext(u, i)$. The predicted context will then be compared to current context of u (that can be inferred). We use a collaborative approach for calculating context of a $(user, item)$ pair. The similarity between two items i and j is computed using the cosine similarity as follows:

$$\text{contexualSimilarity}(i,j) = \frac{\sum_u \text{commonLabels}(i,j)}{\sqrt{\sum_u |\text{labels}(i)| \times |\text{labels}(i)|}} \tag{4}$$

Where commonLabels(i, j) is the number of times users assign the same trip type category to both i and j and labels(i) counts the number of trip type labels given to i by all users. This similarity is used to obtain a neighborhood for item i by selecting the top N most similar items. Then, the predicted context can be computed as in equation 5. In the predicted context, the probability of each trip category is calculated by taking the weighted average of its probabilities in the neighbors' contexts.

$$\text{predictedContext}(u,i) = \frac{\sum_{k \in \text{Neighbors}(i)} \text{context}_u^k \cdot \text{contexualSimilarity}(k,i)}{\sum_{k \in \text{Neighbors}(i)} |\text{contextualSimilarity}(k,i)|} \tag{5}$$

Where context$_u^k$ stands for the neighbor k context given by user u.

Our notion of predicted context for a $(user, item)$ pair is somehow similar to the idea of "best context" introduced in [20] for music recommendation. The authors have defined this concept as the contextual information most suited for a particular item. They have used a vector representation of context where each dimension corresponds to a contextual attribute. if the user believes that context is suitable for that specific item, the value of the corresponding dimension is set to one. They have proposed four different approaches for the prediction of the best context. The first method is based on averaging the context vectors of the item across all users. Another technique is to find the K-nearest neighbors of the user (based on rating history) and compute the predicted context as the weighted average of the contexts assigned to that item by his neighbor. The other two methods follow the same approach except that the similarity of users are computed based on the context vectors and independent of their rating history. Our method is different from the previously mentioned approaches in various aspects: The above methods focus on predicting the suitable context for a $(user, item)$ pair while we address the whole process of context-aware recommendation; In other words, predicting the best context is just a part of our context-aware algorithm. Moreover, our method for calculation of contextual similarity and also prediction of the best context is different from the previous techniques.

Context score of item i for user u can be estimated by comparing the distribution of inferred context of u and predicted context for this item. We used three different methods namely Chebyshev Similarity [21], Kullback-Leibler Similarity [22] and a simple cosine similarity. We have chosen to use cosine similarity in our evaluations as it performs better on our dataset.

Let IC_u denote the inferred context for user u and PC_u^i indicate the predicted context (calculated based on equation 5). The context Score for item i and user u is computed as follows:

$$\text{contextScore}(u,i) = \frac{IC_u \cdot PC_u^i}{||IC_u|| ||PC_u^i||} \tag{6}$$

The utility score of item i and user u is calculated as a function of both the context score of i and also the predicted rating of the item. In our experiments, standard item-based kNN was used to calculate the predicted ratings.

$$\begin{aligned}\text{utility}(i,j) = \alpha \cdot \text{predictedRating}(u,i) + \\ (1-\alpha) \cdot \text{contextScore}(u,i)\end{aligned} \tag{7}$$

In relation 7, α is a constant representing the weight of the predicted rating in the utility function. The items are sorted based on utility values and the top N items are suggested to the user.

4 Evaluation

The evaluations presented in this paper were performed on the Trip Advisor dataset that contains 12558 reviews for 8941 hotels made by 1071 reviewers. About 9500 of the reviews has the "trip type" label which has been used as an indication of context.

Our system consists of two main parts and the experiments have been designed accordingly. The first experiment focuses on assessing the accuracy of the context inference module on our dataset. In the second experiment, the performance of the recommender system is compared with the standard kNN recommender.

4.1 Context Inference Evaluation

The accuracy of the context inference algorithm plays a significant role in the performance of the system. As previously explained, we used Labeled-LDA as it has shown to perform relatively better than other multi-Labeled text classification method. In this experiment we will assess its performance on our dataset. The experiment was set up as a five-fold cross validation. In each of the five runs, one of the folds was used for testing while the topic model was built based on the remaining four folds. For every test case (i.e. the review text), the probability distribution over the trip type categories were predicted. A category is assigned to a test case if the predicted probability for that category exceeds a certain threshold.

The results are evaluated by measuring both precision and recall where precision is computed as the fraction of correct categorical labels, and recall is computed as the ratio of correct labels to total number of labels. Figures 3 and 4 depict recall and precision values for different categories. As

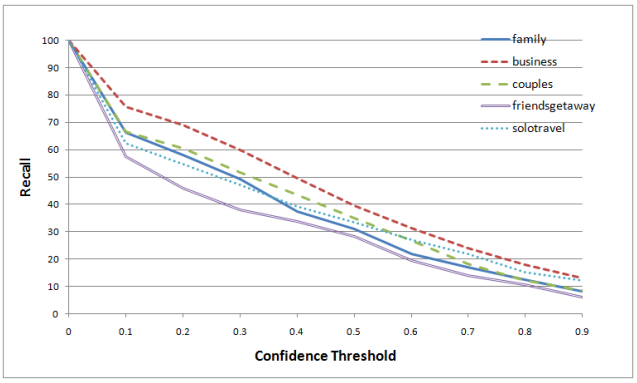

Figure 3: Recall values for different categories

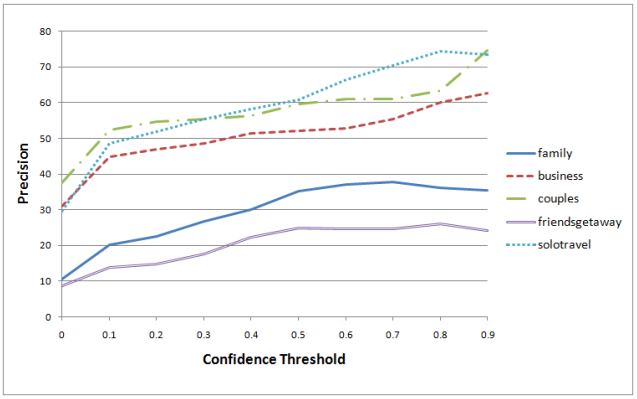

Figure 4: Precision values for different categories

it is shown, the precision tends to be higher as the threshold increases. Also, as expected, by increasing the confidence threshold, recall is likely to decrease.

4.2 Evaluation of Recommendations

As we are working with a sparse dataset, a preprocessing phase has been added to the procedure in order to prune the matrix by removing all those items that have less than 5 ratings.

In previous sections, we introduced a context-aware recommender that produce recommendations for a user based on a utility function that depends both the user's current context and also the predicted rating for that item. As recommendations are based on utility function (and not ratings alone), it is not logical to use metrics such as MAE and other metrics that compare the predicted rating with the actual ones. Instead, hit ratio was chosen as our performance measure and we performed a leave-one-out cross validation experiment on those reviews that have ratings greater than the reviewer's average rating. Having the recommendation size of k, the hit ratio is calculated as the probability that the left-out item is included in the list of N recommendations. The standard item-based kNN algorithm has also been run on the same dataset and under the same condition as our recommender method. Figure 5 shows the hit ratio having different sizes of recommendation list for standard kNN and context aware recommender system where the user's context is inferred and also when it is explicitly expressed. The results suggest that an increase in hit ratio is expected when the contextual information is involved in producing the recommendations.

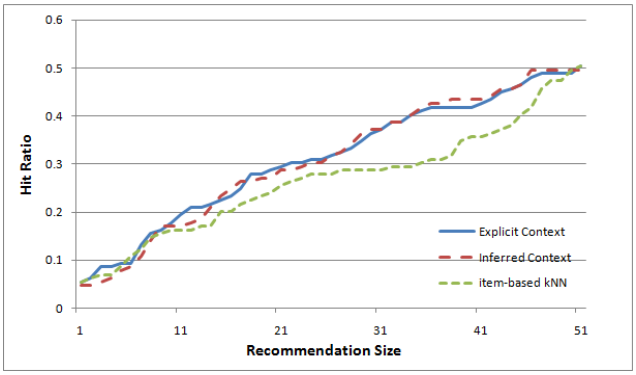

Figure 5: Hit Ratio comparison for item-based kNN and context-aware recommender

5 Conclusions

This paper has presented a novel approach for mining context from unstructured text and using it to produce context-aware recommendations. In our system, the context inference is modeled as a supervised topic-modeling problem for which we used Labeled-LDA to build the context classifier. The inferred context is used to define a utility function for the items reflecting how much each item is preferred by a user given his current context. The utility value for each item depends on two factors: the predicted rating and the "*context score*" where context score represents the suitability of the item for a user in a given context. Rating can be predicted based on any conventional recommendation algorithms such as kNN.

As an example application, we have used our method to mine hidden contextual data from customers' reviews of hotels in "Trip Advisor" dataset and used it to produce context-aware recommendations. Our evaluations indicate that using the contextual information can improve the performance of the recommender system in terms of hit ratio.

References

[1] D. Ramge, D. Hall, R. Nallapati, and C. Manning, "Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[2] G. Abowd, A. Dey, N. D. P.J. Brown, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," *Handheld and Ubiquitous Computing*, vol. 1707, no. 2, pp. 304–307, 1999.

[3] H. Lieberman and T.Selker, "Out of context: Computer systems that adapt to, and lean from, context," *IBM Systems Journal*, vol. 39, no. 3, pp. 617–632, 2000.

[4] W. Woerndl and J. Schlichter, "Introducing context into recommender systems," in *Proceedings of AAAI Workshop on Recommender Systems in E-Commerce*, 2007, pp. 138–140.

[5] P. Dourish, "What do we talk about when we talk about context," *Personal and Ubiquitous Computing*, vol. 8, no. 1, pp. 19–30, 2004.

[6] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Proceedings of the 2008 ACM conference on Recommender Systems*. ACM, 2008.

[7] S. Bourke, K. McCarthy, and B. Smyth, "The social camera: Recommending photo composition using contextual features," in *Proceedings of Workshop on Context-Aware Recommender System*. ACM, 2010.

[8] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, "Developing a context-aware electronic tourist guide: some isues and experiences," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, p. 17.

[9] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices," *Applied Artificial Intelligence*, vol. 17, no. 8, pp. 678–714, 2003.

[10] M. V. Setten, S. Pokraev, and J. Koolwaaij, "Context-aware recommendations in the mobile tourist application compass," in *Proceedings of Third International Conference In Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, August 2004.

[11] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *Proceedings of Workshop on Context-Aware Recommender System*. ACM, 2009.

[12] T. Bogers, "Movie recommendation using random walks over the contextual graph," in *Proceedings of Workshop on Context-Aware Recommender System*. ACM, 2010.

[13] S. Anand and B. Mobasher, "Contextual recommendation," *From Web to Social Web: Discovering and Deploying User and Content Profiles*, 2007.

[14] S. Aciar, "Mining context information from consumers reviews," in *Proceedings of Workshop on Context-Aware Recommender System*. ACM, 2010.

[15] T. Hoffman, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR99)*, 1999.

[16] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, 2003.

[17] D. Blei and J. McAuliffe, "Supervised topic models," *Neural Information Processing Systems*, vol. 21, 2007.

[18] L. Julien, F. Sha, and M. I. Jordan, "Disclda: Discriminative learning for dimensionality reduction and classification," *Neural Information Processing Systems*, vol. 22, 2008.

[19] D. j. S. W. R. Gilks, S. Richardson, *Markov chain Monte Carlo in practice*. London: Chapman & Hall, 1996.

[20] L. Baltrunas, M. Kaminskas, F. Ricci, L. Rokach, B. Shapira, and K. Luke, "Best usage context prediction for music tracks," in *Proceedings of the 2nd Workshop on Context Aware Recommender Systems*, September 2010.

[21] C. Cantrell, *Modern Mathematical Methods for Physicists and Engineers*. Cambridge University Press, 2000.

[22] R. L. S. Kullback, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.