Contextual Recommendation based on Text Mining

Yize Li, Jiazhong Nie, Yi Zhang
School of Engineering
University of California Santa Cruz
{yize,niejiazhong,yiz}@soe.ucsc.edu

Bingqing Wang *
School of Computer Science
Fudan University
wbq@fudan.edu.cn

Baoshi Yan, Fuliang Weng
Research and Technology Center
Robert Bosch LLC
{Baoshi.Yan,Fuliang.Weng}@us.bosch.com

Abstract

The potential benefit of integrating contextual information for recommendation has received much research attention recently, especially with the ever-increasing interest in mobile-based recommendation services. However, context based recommendation research is limited due to the lack of standard evaluation data with contextual information and reliable technology for extracting such information. As a result, there are no widely accepted conclusions on how, when and whether context helps. Additionally, a system often suffers from the so called cold start problem due to the lack of data for training the initial context based recommendation model. This paper proposes a novel solution to address these problems with automated information extraction techniques. We also compare several approaches for utilizing context based on a new data set collected using the proposed solution. The experimental results demonstrate that 1) IE-based techniques can help create a large scale context data with decent quality from online reviews, at least for restaurant recommendations; 2) context helps recommender systems rank items, however, does not help predict user ratings; 3) simply using context to filter items hurts recommendation performance, while a new probabilistic latent relational model we proposed helps.

1 Introduction

In the information retrieval community, one major research focus is developing proactive retrieval agent that acts in anticipation of information needs of a user and recommends information to the user without requiring him/her to issue an explicit query. The most popular examples of such kind of proactive retrieval agent are recommender systems. Over the last several years, research in standard recommender systems has been improved significantly, largely due to the availability of large scale evaluation data sets such as Netflix. One current major research effort is to go beyond the standard user-item rating matrix. As researchers start to realize that the quality of recommendations depends on time, place and a range of other relevant users' context, how to integrate contextual information for recommendation is becoming an ever increasingly important topic in the research agenda (Adomavicius and Ricci, 2009).

One major challenge in context-aware recommendation research is the lack of large scale annotated data set. Ideally, a good research data set should contain contextual information besides users' explicit ratings on items. However, such kinds of data sets are not readily available for researchers. Previous research work in context based recommendation usually experiments on a small data set collected through user studies. Although undoubtedly useful, this approach is limited because 1) user studies are usually very ex-

[*] Bingqing Wang's work is done during his stay in the Research and Technology Center, Robert Bosch LLC

1	I was very excited to try this place and **my wife** took me here on my **birthday**... We ordered a side of the brussell sprouts and they were the highlight of the **night**.
2	**A friend of mine** suggested we meet up here for a **night** of **drinks**... This actually a restaurant with a bar in it, but when we went it was **10pm** and ...

Table 1: Examples of the restaurant reviews

pensive and their scales are small; 2) it is very hard for the research community to repeat such study; and 3) a personalized contextual system may not succeed until a user has interacted with it for a long period of time to enable context based recommendation models well trained.

On the other hand, a large amount of review documents from web sites such as tripadvisor.com, yelp.com, cnet.com, amazon.com, are available with certain contextual information, such as time and companion, implicitly in the reviews (see Table 1 for examples). This offers us an opportunity to apply information extraction techniques for obtaining contextual information from the review texts. Together with users' explicit ratings on items, this might lead to a large research data set for context based recommendation and consequently address the cold start issue in the recommender systems. This paper describes the methods that extract the contextual information from online reviews and their impact on the recommendation quality at different accuracy levels of the extraction methods.

Another challenge is how to integrate contextual information into existing recommendation algorithms. Existing approaches can be classified into three major categories: pre-filtering, post-filtering and the modeling based approaches (Oku et al., 2007; Adomavicius and Tuzhilin, 2008). Pre-filtering approaches utilize contextual information to select data for that context, and then predict ratings using a traditional recommendation method on the selected data (Adomavicius et al., 2005). Post-filtering approaches first predict ratings on the whole data using traditional methods, then use the contextual information to adjust results. Both methods separate contextual information from the rating estimation process and leads to unsatisfying findings. For example, Adomavicious et al. (2005) found neither standard collaborative filtering nor contextual reduction-based methods dominate each other in all the cases. In the modeling based approaches, contextual information is used directly in the rating prediction process. For example, Oku et al. (2007) propose a context-aware SVM-based predictive model to classify restaurants into "positive" and "negative" classes, and contextual information is included as additional input features for the SVM classifier. However, treating recommendation as classification is not a common approach, and does not take advantage of the state of art collaborative filtering techniques. In this paper, we propose a new probabilistic model to integrate contextual information into the state of art factorization based collaborative filtering approach, and compare it with several baselines.

2 Mining Contextual Information from Textual Opinions

The context includes any information that can be used to characterize the situation of entities. Examples of context are: location, identity and state of people, companions, time, activities of the current user, the devices being used etc. (Lee et al., 2005). Without loss of generality, we looked into widely available restaurant review data. More specifically, we investigated four types of contextual information for a dining event, as they might affect users' dining decisions, and they have not been studied carefully before. The four types of contextual information are: *Companion* (whether a dining event involves multiple people), *Occasion* (for what occasions the event is), *Time* (what time during the day) and *Location* (in which city the event happens).

2.1 Text Mining Approaches

We developed a set of algorithms along with existing NLP tools (GATE (Cunningham et al., 2002) etc.) for this task. More detailed description of these algorithms is given below.

Time: we classified the meal time into the following types: "breakfast", "lunch", "dinner", "brunch", "morning tea", "afternoon tea". We compiled a list of lexicons for these different types of meal times, and used a string matching method to find the explicit meal times from reviews. Here, the meal time with an expression, such as "6pm", was extracted using ANNIE's time named entity recognition module from the GATE toolkit. For example, if a user says, "When we went there, it was 10pm", we infer that it was for dinner.

Occasion: The ANNIE's time named entity recognition module recognizes certain special days from text. We augmented ANNIE's lookup function with a list of holidays in the United States from Wikipedia[1] as well as some other occasions, such as birthdays and anniversaries.

Location: Ideally, a location context would be a user's departure location to the selected restaurant. However, such information rarely exists in the review texts. Therefore, we used the location information from a user's profile to approximate.

Companion: Extracting a companion's information accurately from review data is more difficult. We utilized two methods to address the challenge:

Companion-Baseline: This is a string matching based approach. First, we automatically generated a lexicon of different kinds of companion words/phrases by using prepositional patterns, such as "with my (our) NN (NNS)". We extracted the noun or noun phrases from the prepositional phrases as the companion terms, which were then sorted by frequency of occurrence and manually verified. This led to a lexicon of 167 entries. Next, we grouped these entries into 6 main categories of companions: "family", "friend", "couple", "colleague", "food-buddy" and "pet". Finally, the review is tagged as one or more of the companion categories if it contains a corresponding word/phrase in that lexicon.

Companion-Classifier: In order to achieve better precision, we sampled and annotated 1000 sentences with companion terms from the corpus and built three classifiers: 1) a MaxEnt classifier with bag-of-words features, 2) a rule-based classifier, 3) a hybrid classifier. For the rule-based classifier, we looked into the structural aspects of the window where companion terms occurred, specifically, the adjacent verbs and prepositions associated with those terms. We collected high frequency structures including verbs, verb-proposition combinations, and verb-genitive combinations from the whole corpus, and then constructed a list of rules to decide whether a companion context exists based on these structures. For the hybrid classifier, we used the patterns identified by the rule-based classifier as features for the MaxEnt model (Ratnaparkhi, 1998). To train the classifier, we also included features such as POS tags of the verb and of the candidate companion term, the occurrence of a meal term (e.g. "lunch", "dinner"), the occurrence of pronouns (e.g. "we" or "us") and the genitive of the companion term. Based on the evaluation results (using 5-fold cross validation) shown in Table 2, the hybrid classifier is the best performing classifier and it is used for the subsequent experiments in the paper.

	Words	Rule	Hybrid
Precision	0.7181	0.7238	0.7379
Recall	0.8962	0.8947	0.9143
F-Score	0.7973	0.8003	0.8167

Table 2: Evaluation results for the bag-of-words-based classifier (Words), the rule-based classifier (Rule) and the hybrid classifier (Hybrid)

3 Recommendation based on Contextual Information

Next we consider how to integrate various contextual information into recommender systems. Assume there are N items and M users. Each user reviews a set of items in the system. The data set can be represented as a set of quadruplet $D = (y, i, j, \mathbf{c})$, where i is the index of user, j is the index of item, $\mathbf{c}$ is a vector describing the context of this rating data, and y is the rating value. Let $\mathbf{c} = (c_1, ..., c_k)$, where each component c_k represents a type of context, such as "dinner time" or "location=San Jose". The observed features (meta data) of user i and item j are represented as vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ respectively, where each com-

[1] http://en.wikipedia.org/wiki/List_of_holidays_by_country#United_States_of_America

ponent in the vector represents a type of feature, such as "gender of the user" or "price range of the restaurant". In the rest of this paper, we integrate context $\mathbf{c}$ into the user's observed features $\mathbf{f}_i$. This makes $\mathbf{f}_i$ a dynamic feature vector, which will change with different context. The goal is to predict ratings for candidate items given user i and context $\mathbf{c}$, and recommend the top items. We present two recommendation models for integrating contextual information in this section.

3.1 Boolean Model

The Boolean Model filters out items that do not match the context. The Boolean model itself returns an item set instead of a ranked list. We further rank the items by predicted rating values. We score items by the Boolean model as follows:

$$s(j) = \begin{cases} s_m(j) & \text{if item } j \text{ matches the context} \\ -\infty & \text{otherwise} \end{cases} \tag{1}$$

where $s_m(j)$ is the predicted rating computed using a rating prediction method m, such as a Collaborative Filtering model without using context.

3.2 Probabilistic Latent Relational Model

We propose a novel Probabilistic Latent Relational Model (PLRM) for integrating contextual information. In a context-aware recommender system, a user's interest for item is influenced by two factors: (1) the user's long-term preference, which can be learned from users' rating history; (2) the current context (how the item matches the current context). To capture the two factors simultaneously, we introduce a new probabilistic model by assuming the rating value $y_{i,j,\mathbf{c}}$ follows a Gaussian distribution with mean $u_{i,j,\mathbf{c}}$ and variance $1/\lambda^{(y)}$:

$$y_{i,j,\mathbf{c}} \sim \mathcal{N}(u_{i,j,\mathbf{c}}, 1/\lambda^{(y)}) \tag{2}$$

$$u_{i,j,\mathbf{c}} = \mathbf{u}_i^T A \mathbf{v}_j + (W_u \mathbf{f}_i)^T (W_v \mathbf{f}_j) \tag{3}$$

where $\mathbf{u}_i$ and $\mathbf{v}_j$ are the hidden representations of user i and item j to be learned from rating data, and W_u and W_v are feature transformation matrices for users and items respectively. In Equation (3), the first term $\mathbf{u}_i^T A \mathbf{v}_j$ is the estimation based on user' long term preferences, where $A = \{a\}$ is a matrix modeling the interaction between $\mathbf{u}_i$ and $\mathbf{v}_j$.[2] The second term $(W_u \mathbf{f}_i)^T (W_v \mathbf{f}_j)$ is the estimation based on current context and the observed features of users and items, since the context $\mathbf{c}$ is integrated into user's observed features $\mathbf{f}_i$.

$\{U, V, A, W\}$ are the parameters of the model to be estimated from the training data set D, where $W = \{W_u, W_v\} = \{w\}$, $U = \{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_N\}$ and $V = \{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_M\}$. We assume the prior distribution of the parameters follow the Gaussian distributions centered on 0. We use $1/\lambda^{(u)}$,$1/\lambda^{(v)}$, $1/\lambda^{(w)}$ and $1/\lambda^{(a)}$ to represent the variance of the corresponding Gaussian distributions. The effect of the prior distribution is similar to the ridge regression (norm-2 regularizer) commonly used in machine learning algorithms to control model complexity and avoid overfitting.

The proposed model is motivated by well performing recommendation models in the literature. It generalizes several existing models. If we set A to the identity matrix and W_u, W_v to zero matrices, the model presented in Equation (3) is equivalent to the well known norm-2 regularized singular value decomposition, which performs well on the Netflix competition(Salakhutdinov and Mnih, 2007). If we set A to zero matrix and W_u to identity matrix, the Model (3) becomes the bilinear model that works well on Yahoo news recommendation task (Chu and Park, 2009).

Based on the above model assumption, the joint likelihood of all random variables (U, V, A, W and D) in the system is:

$$P(U, V, A, W, D) = \prod_{(i,j,c,y) \in D} P(y_{i,j,c} | \mathbf{u}_i, \mathbf{v}_j, \mathbf{f}_i, \mathbf{f}_j, A, W_u, W_v) \prod_i P(\mathbf{u}_i) \prod_j P(\mathbf{v}_j) P(A) P(W_u) P(W_v) \tag{4}$$

3.3 Parameter Estimation

We use a modified EM algorithm for parameter estimation to find the posterior distribution of (U, V) and max a posterior (MAP) of (A, W).

[2] We introduce A matrix so that the model can also be used to model multiple different types of relationships/interactions jointly, where each type of relationship corresponds to a different A matrix. For the task in this paper, A is not required and can be set to the identity matrix for simplicity. However, we leave A as parameters to be estimated in the rest of this paper for generality.

The estimation can be used to make the final predictions as follows:

$$\hat{y}_{i,j,c} = \int_{\mathbf{u}_i,\mathbf{v}_j} P(\mathbf{u}_i)P(\mathbf{v}_j)(\mathbf{u}_i^T A\mathbf{v}_j + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j)d\mathbf{u}_i d\mathbf{v}_j$$

E Step: the Variational Bayesian approach is used to estimate the posterior distributions of U and V. Assuming (A, W) are known, based on Equation 4, we have

$$P(U,V|A,W,D) \propto \prod_{(y,i,j,\mathbf{c})\in D} \mathcal{N}(\mathbf{u}_i^T A\mathbf{v}_j + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j, 1/\lambda^{(y)}) \times \prod_{i=1}^{M} \mathcal{N}(\mathbf{u}_i|\mathbf{0}, 1/\lambda^{(u)}I) \prod_{j=1}^{N} \mathcal{N}(\mathbf{v}_j|\mathbf{0}, 1/\lambda^{(v)}I)$$

Deriving the exact distribution and use it to predict y will result in intractable integrals. Thus we approximate the posterior with a variational distribution $Q(U,V) = \prod_{i=1}^{M} Q(\mathbf{u}_i)\prod_{j=1}^{N} Q(\mathbf{v}_j)$. $Q(\mathbf{u}_i)$ and $Q(\mathbf{v}_j)$ are restricted to Gaussian distributions so that predicting y using Bayesian inference with $Q(U,V)$ will be straightforward. $Q(U,V)$ can be estimated by minimizing the KL-divergence between it and $P(U,V|A,W,D)$. Since $Q(U,V)$ is factorized into individual $Q(\mathbf{u}_i)$ and $Q(\mathbf{v}_j)$, we can first focus on one $Q(\mathbf{u}_i)$ (or $Q(\mathbf{v}_j)$) at a time by fixing/ignoring other factors. For space considerations, we omit the derivation in this paper. The optimal $Q(\mathbf{u}_i)$ is $\mathcal{N}(\bar{\mathbf{u}}_\mathbf{i}, \mathbf{\Sigma_i})$, where $\bar{\mathbf{u}}_\mathbf{i} = \Sigma_i\mathbf{d}_i$,

$$\begin{aligned}\Sigma_i^{-1} &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)} A(\bar{\mathbf{v}}_\mathbf{j}\bar{\mathbf{v}}_\mathbf{j}^T + \Psi_j)A^T \\ &+ \lambda^{(u)}I \\ \mathbf{d}_i &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)}\tilde{y}A\bar{\mathbf{v}}_\mathbf{j}\end{aligned}$$

Similarly, the optimal $Q(\mathbf{v}_j)$ is $\mathcal{N}(\bar{\mathbf{v}}_\mathbf{j}, \mathbf{\Psi_j})$, where $\bar{\mathbf{v}}_\mathbf{j} = \Psi_j\mathbf{e}_j$,

$$\begin{aligned}\Psi_j^{-1} &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)} A^T(\bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{u}}_\mathbf{i}^T + \Sigma_i)A \\ &+ \lambda^{(v)}I \\ \mathbf{e}_j &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)}\tilde{y}A^T\bar{\mathbf{v}}_\mathbf{j}\end{aligned}$$

Dataset	#Ratings	#Users	#Items
All	756,031	82,892	12,533
Location	583,051	56,026	12,155
Time	229,321	49,748	10,561
Occasion	22,732	12,689	4,135
Companion	196,000	47,545	10,246

Table 3: Statistics of data

M Step: Based on the approximate posterior estimation $Q(U,V)$ derived in the E step, the maximum a posteriori estimation of $\{A, W\}$ can be found by maximizing the expected posterior likelihood $\{\hat{A}, \hat{W}\} = arg\max_{A,W} E_{Q(U,V)}(\log P(A,W,U,V|D))$. This can be done using the conjugate gradient descent method, and the gradient of A, W_u, W_v can be calculated as follows:

$$\begin{aligned}\frac{\partial\Phi}{\partial A} &= \sum_{(y,i,j,c)\in D} \lambda^{(y)}((\hat{y}-y)\bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{v}}_\mathbf{j}^T \\ &+ \bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{u}}_\mathbf{i}^T A\Psi_j + \Sigma_i A\bar{\mathbf{v}}_\mathbf{j}\bar{\mathbf{v}}_\mathbf{j}^T + \Sigma_i A\Psi_j) \\ &+ \lambda^{(a)}A \\ \frac{\partial\Phi}{\partial W_u} &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)}(\hat{y}-y)W_v\mathbf{f}_j\mathbf{f}_i^T \\ &+ \lambda^{(w)}W_u \\ \frac{\partial\Phi}{\partial W_v} &= \sum_{(y,i,j,\mathbf{c})\in D} \lambda^{(y)}(\hat{y}-y)W_u\mathbf{f}_i\mathbf{f}_j^T \\ &+ \lambda^{(w)}W_v\end{aligned}$$

where $\Phi = E_{Q(U,V)}(\log P(A,W,U,V|D))$ and $\hat{y} = \bar{\mathbf{u}}_\mathbf{i}^\mathbf{T} A\bar{\mathbf{v}}_\mathbf{j} + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j$, which is the estimated rating value from user i to item j according to current context and observed features .

4 Experimental Methodology

4.1 Data Collection

We created an evaluation data set based on data collected from a popular review web site where users review services/products and provide integer ratings from 1 to 5. The user profile and the description of items, such as user gender and the category of restaurants are also collected. The data set used in this paper includes the restaurants in Silicon Valley (Bay area) and the users who ever reviewed these restaurants. We extract context from the review texts. The four kinds of context considered in our paper are described in Section 2.1. For each type of context, we create a subset, in which all reviews contain the corresponding contextual information. Finally we construct four sub data sets and each data set is described by the corresponding context type: Time, Location, Occasion and Companion. We use “All” to represent the whole data set. Statistics about each data set are described in Table 3.

4.2 Experimental Setup

We design the experiments to answer the following questions: 1) Does including contextual information improve the recommendation performance? 2) How does the probabilistic latent relational modeling approach compare with pre-filtering or post-filtering approaches? 3) How does the extraction quality of the contextual information affect the recommendation performance?

To answer the first question, we compare the performance of the Probabilistic Latent Relational Model on a standard collaborative filtering setting where only rating information is considered, indicated by Nocontext. We also evaluate the performance of the Probabilistic Latent Relational Model when integrating contextual information, indicated by Context-X, where X represents the type of contextual information considered. To answer the second question, we compare the performance of Context-X with the pre-filtering Boolean Model, which first uses the context to select items and then ranks them using scores computed by Nocontext. To answer the third question, we compare the recommendation performance for different extraction precision. The performance on the following two recommendation tasks are reported in this paper:

Top-K Recommendation: We rank the items by the predicted rating values and retrieve the top K items. This task simulates the scenario where a real recommender system usually suggests a list of ranked K items to a user. To simulate the scenario that we only want to recommend the 5-star items to users, we treat 5-star rating data in testing data as relevant. Ideally, classic IR measures such as Precision and Recall are used to evaluate the recommendation algorithms. However, without complete relevance judgements, standard IR evaluation is almost infeasible. Thus we use a variation of the evaluation method proposed by Koren (Koren, 2008).

Rating Prediction: Given an active user i and a target item j, the system predicts the rating of user i on item j. The prediction accuracy is measured by Root Mean Square Error (RMSE), which is commonly used in collaborative filtering research. This task simulates the scenario that we need to guess a user’s rating about an item, given that the user has already purchased/selected the item.

For each data set (Time, Companion, Location, Occasion and All), we randomly sample 10% for testing, 80% for training and 10% for validation.

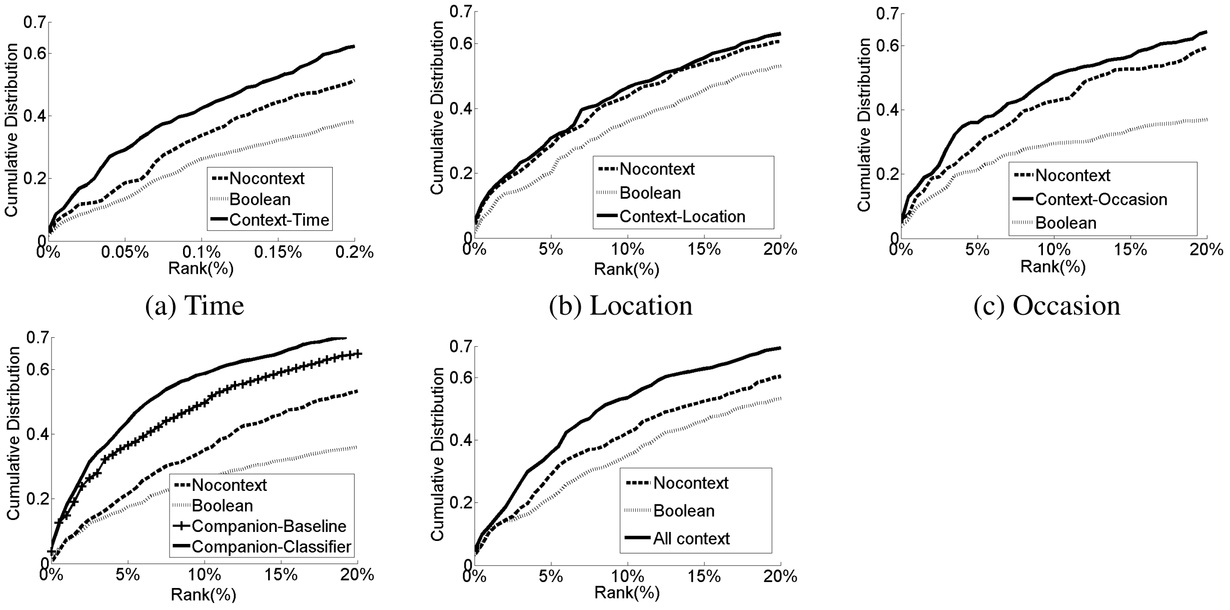
5 Experimental Results

5.1 Performance on Top-K Recommendation

Figure 1(a)-(e) shows the ranking performance on each data set. The x-axis is the rank and the y-axis is the portion of relevant products covered by this level of rank. The results across all data sets are consistent. With contextual information, PLRM Context-X outperforms Nocontext, whereas using context to pre-filter items (Boolean) does not help. It means that contextual information can help if used appropriately, however improperly utilizing context, such as simply using it as a boolean filter, may hurt the recommendation performance. Our proposed PLRM is an effective way to integrate contextual information.

5.2 Performance on Rating Prediction Task

Table 4 summaries the RMSE results of different approaches on the rating prediction task. The RMSE of simply using item’s average rating value as the prediction is also reported as a reference since it is a commonly used approach by non personalized recommender systems. For each context, we can either train the model only on the subset that consists of rating data with related context, or train on a bigger data set by adding the rating data without related context. The results on both settings are reported here. Table 4 shows that utilizing context does not affect the prediction accuracy. We may wonder why the effects of adding context is so different on the rating task compared



(a) Time

(b) Location

(c) Occasion

(d) Companion

(e) All

Figure 1: Performance on the top-K recommendation task. The plots focus on the top 20% ranking region.

	Training on Sub Data set			Training on the Whole Data set		
Testing Data	ItemAvg	Nocontext	Context	ItemAvg	Nocontext	Context
Time	1.1517	1.0067	1.0067	1.1052	0.9829	0.9822
Companion	1.2657	1.0891	1.0888	1.2012	1.0693	1.0695
Occasion	1.2803	1.1381	1.1355	1.2121	1.0586	1.0583
Location	1.1597	1.0209	1.0206	1.1597	1.0183	1.0183
All context	-	-	-	1.1640	1.0222	1.0219

Table 4: RMSE on the rating prediction task

	Time	CompanionBaseline	CompanionClassifier	Occasion
#Reviews	300	300	300	200
#Contexts	115	148	114	207
Precision	84.4%	62.2%	77.1%	-
Recall	80.2%	95.8%	91.7%	-
F-Score	82.2%	75.4%	83.8%	Accuracy 78.3%

Table 5: Performance of the context extraction module

with the ranking task. One possible explanation is that the selection process of a user is influenced by context, while how the user rates an item after selecting it is less relevant to context. For example, when a user wants to have a breakfast, he may prefer a cafeteria rather than a formal restaurant. However, how the user rates this cafeteria is more based on user's experiences in the cafeteria, such as quality of services, food, price, environment, etc.

5.3 How does Text Mining Accuracy Affect Recommendation

To evaluate the extraction performance on "Companion", "Time" and "Occasion", we randomly sample some reviews and evaluate the performance on the samples[3]. The results are shown in Table 5. Compared with other contexts, the extraction of companion context is more challenging and the string matching baseline algorithm produces significantly inferior results. However, by using a MaxEnt classifier with features selection, we can boost the precision of the companion context extraction to a level comparable to other contexts.

To further investigate the relationship between the quality of the extracted context and the performance of the recommender system, we compare the recommendation performance of Companion-Baseline and Companion-Classifier in Figure 1(d). It shows that improving the quality of the extraction task leads to a significant improvement on the recommender systems' top-K ranking task.

6 Conclusions

Reviews widely available online contain a large amount of contextual information. This paper proposes to leverage information extraction techniques to help recommender systems to train better context-aware recommendation models by mining reviews. We also introduce a probabilistic latent relation model for integrating the current context and the user's long term preferences. This model takes the advantages of traditional collaborative filtering approaches (CF). It also captures the interaction between contextual information and item characteristics. The experimental results demonstrate that context is an important factor that affects user choices. If properly used, contextual information helps ranking based recommendation systems, probably because context influences users' purchasing decisions. Besides, more accurate contextual information leads to better recommendation models. However, contextual information does not help the user rating prediction task significantly, probably because context doesn't matter much given the user has already chosen a restaurant.

As the first step towards using the information extraction techniques to help contextual recommendation, the techniques used in this paper are far from optimal. In the future, we will research more effective text mining techniques for contextual extraction(Mazur and Dale, 2008; McCallum et al., 2000; Lafferty et al., 2001) at the same time increasing the amount of annotated review data for better classifier performance through actively learning (Laws and Schütze, 2008). We also plan to work towards a better understanding of contextual information in recommender systems, and explore other types of contextual information in different types of recommendation tasks besides

[3] We sample 300 reviews for "Time" and "Companion" evaluation. Due to the extremely low probability of occurrence of Occasion context, we futher sample 200 reviews containing Occasion-related expressions and only evaluate extraction accuracy on these samples

restaurant recommendations.

7 Acknowledgements

Part of this research is funded by National Science Foundation IIS-0713111 and the Institute of Education Science. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors. Bingqing Wang's work is done during his stay in the Research and Technology Center, Robert Bosch LLC.

References

Adomavicius, Gediminas and Francesco Ricci. 2009. Recsys'09 workshop 3: workshop on context-aware recommender systems, cars-2009. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009*, pages 423–424.

Adomavicius, Gediminas and Alexander Tuzhilin. 2008. Context-aware recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 335–336.

Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145.

Chu, Wei and Seung-Taek Park. 2009. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, pages 691–700.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Koren, Yehuda. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008*, pages 426–434.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001*, pages 282–289.

Laws, Florian and Hinrich Schütze. 2008. Stopping criteria for active learning of named entity recognition. In *COLING 2008, 22nd International Conference on Computational Linguistics*, pages 465–472.

Lee, Hong Joo, Joon Yeon Choi, and Sung Joo Park. 2005. Context-aware recommendations on the mobile web. In *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, pages 142–151.

Mazur, Pawel and Robert Dale. 2008. What's the date? high accuracy interpretation of weekday names. In *Proceedings of the 22nd International Conference on Computational Linguistics, Coling 2008*, pages 553–560, Manchester, UK, August.

McCallum, Andrew, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning, ICML 2000*, pages 591–598.

Oku, Kenta, Shinsuke Nakajima, Jun Miyazaki, and Shunsuke Uemura. 2007. Investigation for designing of context-aware recommendation system using svm. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2007, IMECS 2007*, pages 970–975.

Ratnaparkhi, A. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Salakhutdinov, Ruslan and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of Neural Information Processing Systems (NIPS)*.