Finding Contexual Preferrable Products from Online Reviews

Minyi Cai, Chen Lin*

School of Information Science and Engineering, Xiamen University, Xiamen, 361005, China

Jian Pei

School of Computing Science, Simon Fraser University, Burnaby, V5A1S6, Canada

[*] corresponding author
Email address: `chenlin@xmu.edu.cn` (Chen Lin)

Finding Contexual Preferrable Products from Online Reviews

Minyi Cai, Chen Lin*

School of Information Science and Engineering, Xiamen University, Xiamen, 361005, China

Jian Pei

School of Computing Science, Simon Fraser University, Burnaby, V5A1S6, Canada

Abstract

In this paper we study the problem of **r**eview **g**uided **c**ontext **a**ware product **r**ecommendation **s**ystems (RGCARS). We propose a framework which is based on utility surplus theory, to asemble the ratings, opinions and contents of online reviews, for the learning of contexual user preferences. The major challenge in RGCARS is that we are facing more complicated data than in traditional recommender systems, i.e. the explicit feedback is not explicitly annotated, and there is a greater variety of implicit feedback which can not be readily dealt with by existing methods. To fully exploit the value of online reviews, including explicit feedback and implicit feedback, we prensent a semi-supervised classifier to automatically identify explicit contextual feedback, and several models respectfully under the assumptions that contextual feedback is fully observable, or partially observable and missing not at random. We conduct comprehensive experiments on real data sets, revealing typical recommendation scenarios such as cold start recommendations, context aware recommendation, visuallization and so on. Experimental results demonstrate that our models outperform state-of-the-art methods.

Keywords: context aware recommendation systems, online review mining, missing not at random, probabilisitc graphic model, utility surplus

*corresponding author
Email address: chenlin@xmu.edu.cn (Chen Lin)

1. Introduction

Product recommender is an indispensible component of modern E-commerce ecosystems. Delivering accurate recommendations benefits both enterprises and consumers. Precise recommendation is built upon fine-tuned modeling of user preferences. As we will soon see in the following example, user preferences are often affected by the "contexts", i.e. when, where or with whom the consumptions are made. Thus, a context independent recommender will make wrong predictions.

Example: *Bob is looking for a place where he can have his weekday lunch alone. A sandwich combo seems to be a perfect option. However, when Bob is preparing for a date with his girlfriend at Christmas' Eve, recommending a fast food restaurant would be inappropriate. Instead, a nice restaurants serving French cuisine and fine wine is more attracting.*

Context **A**ware **R**ecommendation **S**ystems (CARS) take contextual information into account to provide better recommendations. CARS have been successfully applied in a few areas, including travel [6], music [8], web news [31] and so on. As more and more people use review sites to share shopping experiences, recently a few researchers explore the feasibility of **R**eview **G**uided **C**ontext **A**ware **R**ecommendation **S**ystems (RGCARS), i.e. learning context aware user prefernces from online reviews [17, 15, 11, 19].

Naturally, one key problem in RGCARS is to integrate review contents and ratings into the recommender framework. Online reviews do not only provide context aware user feedback, if any, but also act as complementary product profiles. Moreover, many online reviews reveal strong and direct associations between certain product features and context aware preferences. As shown in Tab. 1, different users publish opinions based on their experiences under certain contexts (highlighted in blue). In order to explain their opinions, they usually talk about several aspects of a restaurant (highlighted in red) in detail. The descriptions are subjective and highly dynamic, thus previous researches implementing a static content model [11] or a a collaborative filtering model [17] lack the theoretical foundation to connect the dots among products, user preferences, contexts, purchases and reviews.

To address this problem, we propose a recommender framework, which is inspired by utility surplus theory. Utility is an economic concept to measure consumer's "satisfaction" or "pleasure". It is the aggregation of user preferences over several commodity features of interest. We assume that when a consumer is pleasant, he/she will post a piece of positive review; otherwise, he/she will post

Table 1: Examples of typical online reviews from www.yelp.ca, ratings in range of $[0, 5]$

Type	Example Review	Rating
Explicit negative feedback for context "dating"	On a random Tuesday night, my fiance invited me to a dinner date. For a place who's decor screams relaxation, we felt awfully rushed the whole time.	1
Explicit positive feedback for context "dating"	We also came on a Friday night and were able to be seated right away. This place had great lighting, it was very cozy, still felt very familyesk but would be a wonderful wonderful date spot too.	5
Implicit feedback for context "dating"	I went here with my boyfriend for dinner on a Saturday night. Food was ok. I might consider returning, but not for a while at least.	3
Implicit feedback for any context	On the bar, floors, and walls, finally, the service. There was nothing above and beyond but nothing overly flawed.	4

a negative piece. Therefore, given the product descriptions and the opinions of online reviews, the contextual user preferences can be inferred.

Another issue which, to the best of our knowledge, has not yet been studied by previous RGCARS, is the explicit and implicit feedback in online reviews. In traditional recommender systems, ratings are explicit feedback, while others, such as clicks and the "missing ratings" are regarded as implicit feedback. In the setting of RGCARS, contextual explicit feedback is not exactly "explicit", as they have to be identified with skills whether or not they are relevant to a context. Also, there are two types of implicit feedback, one is the absence of reviews, the other is the absence of context aware reviews. The former is equivalent to the "missing ratings" in traditional recommender systems and has been handled by various means, including one class collaborative filtering [28], binary implicit feedback [26], and missing not at random models [32]. The latter is unique and cannot be readily delt with by previous efforts. As shown in Tab. 1, implicit contextual feedback could be either a comment with neutral opinion on the given context, or a comment not related to any context.

In this paper, we ask questions such as "How do we automatically discover explicit feedback" and "How do we deal with implicit feedback"? Ignoring implicit feedback as missing instances is intuitively not optimal as it may distort inferences about the user preferences. For example, if users are less prone to write explicit contextual reviews for products they dislike, a model purely based on explicit feedback may be biased to inaccurately higher predictions, even for unpreferred items. Our solution is a classifier to obtain explicit feedback, and two distinc approaches to utilize implicit feedback. In the first approach, we treat all implicit feedback as observations, and study how to transform them into postive and negative feedback to reduce the noise introduced in this process. In the second approach, we assume implicit feedback is missing observations, while they are missing not at random (MNAR). We testify these assumptions by modeling that the probability of writing an explicit review is dependent on the contextual utility alone, or the comparison of contextual utility and context-free unitility of the product for the user.

The contributions of this paper are three-fold. (1) We present a unified framework to fully exploit the value of online reviews, including explicit and implicit contextual feedback, for context aware product recommendation. The framework gives economic interpretatation for producing the explicit feedback, as well as observing the feedback. (2) Due to the high volume of online reviews, it is impossible to manually label all context related reviews. We propose a semi-supervised classifier to learn the hidden contexts from a few roughly labeled instances and a large number of unlabeled instances. It helps to clarify ambiguous expressions and to identify the true contexts. (3) We perform comprehensive experiments on real data sets. We apply and extend our models to many typical ecommerce scenarios, such as cold-start recommendation, context-aware recommendation and visualization. Experimental results show that the proposed framework outperforms state-of-the-art methods with or without text mining techniques.

The rest of this paper is organized as follows. We provide a brief discussion on related work in section 2. We introduce the overall framework in Sec. 3, and its components (contextual opinion classification model, utility model with complete observations or missing observations) in section 4, 5 and 6, respectfully. The experimental results are described and analyzed in section 7. Finally, we conclude our work in section 8.

2. Related Work

RGCARS are genetically related to two areas: contextual aware recommendation systems and online review mining.

2.1. Context Aware Recommendation Systems

Most recommender systems use numeric user ratings to construct a rating matrix, and apply collaborative filtering approaches to recommend items for users with similar tastes [7]. There are two types of collaborative filtering techniques. One is based on retrieving nearest neighbors, performances of which are enhanced by modifying the similarity measurements, such as removing the global effect of nearest neighbors [5]. The other is based on matrix factorization [14] to approximate the observed ratings with hidden user preferences and item features. Probabilistic versions of matrix factorization are also prodominant [27]. For binary relevance data, researchers present methodologies that target different object functions, most of which are ranking related measures. For example, BPR [26] maximizes the likelihood of pair-wise rankings, CliMF [28] directly maximizes the mean reciprocal rank to improve top-k recommendations, ListRank-MF [29] minimizes a loss function that represents the uncertainty between training lists and output lists.

Recently, constext aware recommender systems (CARS) have attracted both academic and industrial attentions [2]. The performance of CARS has been verified by a live controlled experiment [9]. There are three types of approaches to incorporate contextual information in the recommendation process. The first type is to pre-filter, i.e. select contextualized ratings data and factorize each context specific rating matrix [1]. The second type is to post-filter, i.e. split the resulting items to different contexts after recommendation [4]. The third type is to model the context, i.e. as a latent variable in BNN [22], or as a latent factor in matrix factorization [3]. or as tensor factorization [31, 13]. Empirical study has shown that which approach is better depends on the application [23]. The sparsity of rating data is an obstacle for CARS. A typical improvement is to integrate other resources, i.e. demographic information [16], sequential patterns [10], or, as we might review in the next subsection, texts in online reviews [17, 15, 11, 19].

2.2. Online Review Mining

Online review mining has been an active research area. Most existing researches are efforts that summarize reviews and extract certain information, i.e. opinion polarities [18], user groups according to their interests [30], aspects of products [20], and so on. Online review mining often requires a skillful combination of natural language processing (NLP) and machine learning models. An omnipotent model does not exist for every domain.

Information extracted from online reviews is helpful in recommender systems. For example, identifying product aspects and user opinions is crucial for predicting a user's rating [24], estimating the review quality can "up-weight" or "down-weight" the importance of individual rating while performing collaborative filtering [25]. For CARS, online review mining also assists the recommendation process in POI recommender [6], hotel recommender [15], and restaurant recommender[17], etc..

For RGCARS, most researches in literature directly utilize the extracted conextual opinions to form preference data, and then pipe the explicit feedback with a CARS model. For example, a tensor factorization model is presented in [15], which imitates a user who favors reviews written by people with the same intent, nationality and tasts. An extended LDA is presented in [11], which jointly models users, items and contexts. In [17], the contextural information is integrated into a probabilistic latent relational model, which factorizes ratings to item specific features, as well as a combination of the current context and a user's long term preference. In [19] a simple recommendation model is used to aggregate opinions over each product feature.

3. Methodology Overview

We consider the most common form of consumer reviews on popular e-commerce sites, such as Amazon, and social media sites, such as TripAdvisor and Yelp. The consumer reviews are in the format of a short passage of texts accompanied by a numerical rating. Suppose in the pre-processing, we extract a predefined set of contextual factors $F = \{f\}$, which can be any textual phrases in natural languages, i.e. "*with my girlfriend*", "*at Christmas Eve*", etc. We also identify the commodity features $C = \{c\}$, i.e. "*good food*", "*wonderful staff*", etc. The associated opinions for the contextual factors and commodity features are denoted by $O = \{o\}$, as listed in Tab. 2. The detailes of pre-processing are described in Sec. 7. We present a castcade framework which sequentially solves the following two problems.

Contextual Opinion Categorization Suppose there are a set of reviews $E = \{e\}$, where each review consists of the user's opinion on the corresponding contextual factors and commodity features, a few reviews are labeled by a set of K conext labels $L = \{l_k\}, 1 \leq k \leq K$, predict if a review is relevant to any context. If so, the review is an explicit response, $r = 1$, predict whether the opinion for the context x is positive or negative. Since there are only a few lableled instances available, the multi-class problem requires a semi-supervision approach.

Context Aware Recommendation Given a set of explicit and implicit contextual review feedback of a group of users $U = \{u\}$ on a universe of commodities

$V = \{v\}$, estimate how a user will like an item under a context.

In model based collaborative filtering, i.e. matrix factorization, user feedback is usually determined by latent variables for user preferences U and item features V. In RGCARS, an additional hidden variable a is introduced for each context. If all the reviews are explicit feedback, i.e. $\forall r, r = 1$, the problem is tackled by maximizing the likelihood of generating the reviews $p(X|A, U, V)$. We present a utility model for complete observations in Sec. 5.

However, if implicit feedback is incorporated, i.e. $\exists r, r = 0$, then the opinions in implicit feedback is missing $X^{mis} \neq \emptyset$, the likelihood is defined as follows.

$$p(X^{obs}, R|A, U, V) = \int_{X^{mis}} p(X|A, U, V) p(R|X, A, U, V) \tag{1}$$

When R is independent given X, $p(X^{obs}, R|A, U, V) \propto p(X^{obs}|A, U, V)$, it degrades to the utility model for complete observations. When R and X are dependent, a model is needed to stimulate the process of generating R given X. We present two variants of the utility model for missing not at random observations in Sec. 6.

4. Opinion Classification

Suppose the labeled reviews can be grouped into $K+1$ categories: $\{E_1, \cdots, E_K, E_{K+1}\}$, where E_k represents reviews related to context k, the last category E_{K+1} consists of reviews that doesn't belong to any context. A softmax [12] regression for this multi-class categorization problem is to minimize:

$$L(\mathbf{b}) = -\Sigma_k \Sigma_{e \in E_k^+} \log \frac{\exp \mathbf{b_k f_e}}{\Sigma_{k'} \exp \mathbf{b_{k'} f_e}} \tag{2}$$

where $\mathbf{f_e}$ is the vector denotion of contextual factors in a review e. In the prediction, we assign a $K-$dimensional label $\mathbf{l}$ for each instance, the context distribution for an unlabeled review $\mathbf{f_e}$ is then computed by $p(l_{e,k} = 1|\mathbf{f_e}) = \frac{\exp \mathbf{b}_k \mathbf{f_e}^T}{\Sigma_{k'} \exp \mathbf{b}_{k'} \mathbf{f_e}^T}$

As mentioned in previous sections, the size of online reviews is too large for complete human annotations. It is critical to use characteristics of online reviews to facilitate contextual label prediction. We have the following assumptions.

1. **Selection Bias** We argue that selection bias exists in consumers purchasing a product. For example, fast food restaurants are more often chosen for casual lunch, and luxe restaurants are reserved for major celebrations. As reviews are associated to commodities, the prior distribution of contexts for

Table 2: List of notations

Variable	Notation and Value	Explanation
Hidden Variables	$x_{u,v,k} \in \{0,1\}$	negative and positive opinion for a user u purchasing a commodity v under a context k
	$y_{u,v,k} \in \{0,1\}$	negative and positive switch for a user u on a commodity v under a context k
	$l_{u,v,k} \in \{0,1\}$[1]	relateness of a review by u on v to context k
Parameters	$u_c \in R$	user u's preference over commodity feature c
	$a_{k,c} \in R$	context aware preference for context k over commodity feature c
	$\mathbf{\Theta}$	probability distribution over feature vocabulary
	$v_c \in (0,1)$	item specific probability for v on feature c
	$\mathbf{b_k} \in R^{\|F\|}$	weights over contextual factors for context k
	$\mathbf{\Pi}$	response probability distribution for explicit or implicit feedback x
Observed variables	E_k^+, E_k^-, E_k^{mis}	set of positive, negative and implicit reviews
	$F = \{f\}$	vocabulary of contextual factors
	$C = \{c\}$	vocabulary of commodity features
	$o_{u,v,c} \in \{0,1\}$	opinion polarity of user u on item v for feature c
	$r_{u,v,k} \in \{0,1\}$	response variable to indicate if a review is explicit feedback
	$s_{u,v} \in [0,1]$	normalized rating by u on v

each commodity is not uniform. If we randomly sample reviews to annotate, then the context distribution of a commodity in the sample set is an approximate estimate of the true context distribution. For example, if the commodity has a context distribution $P\tilde{X}_v = \{$“party:”0.1,“banquet:”0.8,“weekday lunch:”0.1$\}$ in the sample set, then we expect the context “banquet” is also dominant in the unlabeled reviews.

2. **User Activity** Surveys on social media usage often report that consumers are willing to share their opinions (either prises or complaints). We claim that the user activity, which is the portion of context related reviews, is controlled by consumer demographics and irrelevant to the classifier. For example, the prior distribution of context and non-context related reviews, denoted as $\tilde{PU}$, is shared among all reviews and would be expected to stay stable for unlabeled reviews.

We construct $\tilde{P}_v$ as a $K+1$-dimensional vector for each item v.

$$\tilde{P_{v,k}} = \begin{cases} P\tilde{U}_1 P\tilde{X}^{v,k} & 1 \leq k \leq K \\ P\tilde{U}_2 & k = K+1 \end{cases} \tag{3}$$

Following the above assumptions, we devise the KL-divergence between distribution of review samples and the estimated distribution in unlabeled reviews $p^c(\mathbf{l}|\mathbf{b})$ based on the parameter $\mathbf{b}$, which is $D(\tilde{p}\|p(l|\mathbf{b})) = \Sigma_c D_{KL}(\tilde{p^c}\|p^c(l|\mathbf{b}))$ Given a set of labeled reviews E on commodities $C^+ = \{c\}$, and a large number of unlabeled reviews EU, in order to learn the classifier parameters $\mathbf{b}$, we first compute the prior distribution $\tilde{p^c}$ by equation 3 from $\tilde{PU}$ and $P\tilde{X}$, and then optimize the following loss function.

$$\min_{\mathbf{b}} - \frac{1}{E^+} \text{log} p(E^+|\mathbf{b}) + \lambda_p D(\tilde{p}\|p(l|\mathbf{b})) + \lambda_c \|\mathbf{b}\|_2 \tag{4}$$

where the first term is to maximize the likelihood for generating the labeled reviews; the second term is to maximize the similarity between context distribution from sampled annotations and unlabeled data; the third term is a rugularizer to avoid over-fitting.

In the pre-processing, we adopt a dictionary based procedure to identify opinionated phrases and their orientations, and pair them with the closest commodity features and contextual factors. For context related reviews, i.e. labeled by $\exists k, l_k = 1$, we denote the set of positive reviews as E_k^+, negative reviews as E_k^-. Conventional collaborative filtering use biased ratings instead of the absolute value of ratings to indicate preferences. Here we borrow the idea and compare the rating $s_{u,v}$ to a Bayesian estimate of the user specific average rating $\tilde{s_u}$.

$$\tilde{s_u} = \frac{\bar{s_u} \times N_u + \bar{s} \times N}{N + N_u} \tag{5}$$

where $\bar{s_u}$ is the empirical average rating for user u, N_u is the number of ratings for u, $\bar{s}$ is the average rating for all users. N is the minimum number of ratings required to calculate the Bayesian estimate for $\tilde{s_u}$. If $N_u < N$, $\tilde{s_u} = \bar{s}$. If the target rating $s_{u,v} > \tilde{s_u}$, the review is included in E_k^+, otherwise it will be put into E_k^-.

5. Utility Model with Complete Observations

5.1. Utility Surplus

In Economics, utility is an important property of any commodity. It measures the satisfaction consumers get by purchasing an item. Money, as a special case of commodity, can also be measured by a utility function. If the consumer is willing to pay a certain amount of money, which is the price v_p, to purchase an item v, then the utility surplus $US(u,v) = UC(u,v) - UM(u_p, v_p)$, which is the commodity utility UC minus the money utility UM, will be positive.

Usually, the commodity utility can not be directly counted, but can be inferred from observed consumptions. Moreover, a common assumption is that the commodity utility can be modeled as a linear combination of user preferences over commodity features. We extend this theory to the CARS problem. Suppose the user preference is represented by u, and contextual preference is represented by a, then the commodity utility under context k is measured by a combination of user specifict utility and context aware utility: $UC_k(u,v) = \sum_c (\alpha a_{k,c} + (1-\alpha) u_c) v_c$, where $a_{k,c}$ is the contextual preference under context k to the commodity feature c, u_c is the user preference to c, and v_c is the quality of item v on feature c.

The utility function of money could be more complicated. However, in this scenario, the consumers are paying a very small portion of money, compared to their incomes. So according to the law of diminishing marginal utility, a linear function can be adopted to measure the decrease of consumer satisfaction by losing the money, i.e. $UM(u_p, v_P) = u_P v_P$. Here u_P can also be interpreted as users' sensitiveness to price.

We make the following manipulation to ensure that the score coincides with a probability.

$$p(x_{u,v,k} = 1|\mathbf{u}, \mathbf{a}, \mathbf{v}) = g(UC_k(\mathbf{u}, \mathbf{v})) = \frac{1}{1 + \exp -UC_k(\mathbf{u}, \mathbf{v})} \tag{6}$$

The economic interpretation is clear. If the user is satisfied with the consumption for the context, $US_k(u, v) > 0$, then $p(c|x) > 0.5$, which suggests that it is possible to choose the write a positive review.

5.2. Basic Model

Intuitively, if a product is strongly associated with a particular feature, i.e. a restaurant is famous for its dessert, then consumers are more likely to have positive opinions about that feature. Therefore it is naturally to model the probability of user praising a product feature by v. As shown in 1(a), in the basic model, we present the following generation process.

- Generate user preference from a multivariate Gaussian distribution $u \sim \mathcal{N}(u; 0, \sigma_u^2)$, contextual preference $a_k \sim \mathcal{N}(a_k; 0, \sigma_a^2)$, item specific feature strength $v_c \sim Beta(v_c; \zeta_a, \zeta_b)$ for each user, item-feature, and context respectfully. Generate global vocabulary distribution over the feature phrases $\theta \sim Dirichlet(\theta; \xi)$.
- For each review written by u on v under context k
  - Publish a positive review $x \sim Bern(x; g(UC_k(u, v)))$
  - For each product feature
    - Choose feature phrases $c \sim Discrete(c; \theta)$
    - Select an opinion polarity to describe the feature $o \sim Bern(o; v_c)$

The parameters $\mathbf{u}, \mathbf{v}, \mathbf{a}$ are obtained by maximizing the posterior function $\log L(\mathbf{U}, \mathbf{V}, \mathbf{a})$ in Equ.7.

$$L(U, V, a) = p(U, V, a|X, O, C, \sigma_u, \sigma_a, \zeta_a, \zeta_b, \xi) \propto p(X, O, C, U, V, a|\sigma_u, \sigma_a, \zeta_a, \zeta_b, \xi) \tag{7}$$

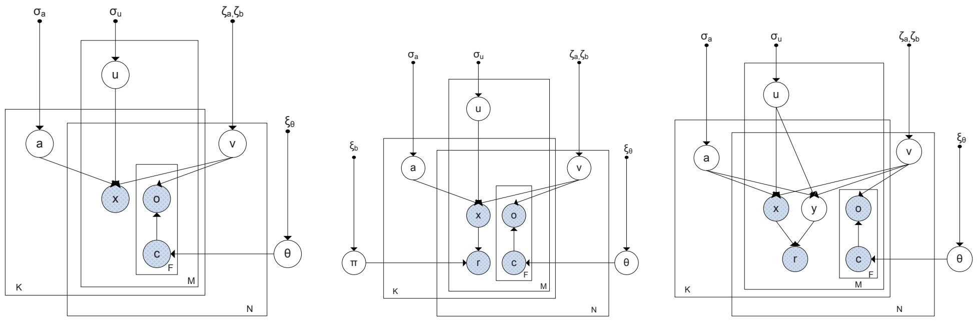
Reviews which have been identified as positive or negative feedback on context k in Sec 4 have clear meanings. Therefore, the “absolute” baseline is to use the recognized explicit reviews with predicted label $l_{u,v,k} = 1$ as training instances, i.e. $x_{u,v,k} = 1$ for reviews in E_k^+ with higher than average ratings,

$x_{u,v,k} = 0$ for reviews in E_k^- with low ratings. We utilize steepest descent in each update step in the inference algorithm 1

In the above "absolute" baseline, implicit feedback with label $l_{u,v,k} = 0$ is not considered. As mentioned in Sec. 1, implicit feedback contains context free reviews and neutral opinions. Our biased rating mechanism in Sec. 4 has the advantage of transforming neutral opinions to binary preferential data. For a context irrelevant review, a "fuzzy" alternate is to treat them as possible training instances with the possibility of being a positive explicit feedback $p(x_{u,v,k} = 1) = l(\mathbf{b})$ computed by the prediction equation Equ. 2. To sum up, the "fuzzy" strategy assigns a weight $l_{u,v,k}$ to each review. For an annotated review related to context k, $l_{u,v,k} = 1$. For a test instance $l_{u,v,k} = \frac{\exp \mathbf{b_k f_e}}{\Sigma_{k'} \exp \mathbf{b_{k'} f_e}}$.

Algorithm 1: Learning Process for model basic

```
Input: A set of positive and negative feedback $\mathbf{x}$, a set of feature opinion
       pairs $\{< c, o >\}$
Output: $\mathbf{a}, \mathbf{V}, \mathbf{U}$
Initialize $\mathbf{a}, \mathbf{V}, \mathbf{U}$;
while not convergent do
    for $x_{u,v,k} \in \mathbf{x}$ do
        $t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} = l_{u,v,k} \frac{\{-\exp[(-\alpha \mathbf{a_k} - (1-\alpha)\mathbf{u})\mathbf{v}]\}^{x_{u,v,k}} 1^{1-x_{u,v,k}}}{1+\exp[(-\alpha \mathbf{a_k} - (1-\alpha)\mathbf{u})\mathbf{v}]}$;
    end
    for $\mathbf{u} \in \mathbf{U}$ do
        $\mathbf{u} = \mathbf{u} - \lambda_t[-(1-\alpha)\Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}} t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} \mathbf{v} - \frac{\mathbf{u}}{\pi\sigma_u^2}]$;
    end
    for $\mathbf{v} \in \mathbf{V}$ do
        $\mathbf{v} = \mathbf{v} - \lambda_t\{\Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}}[-\alpha \mathbf{a_k} - (1-\alpha)\mathbf{u}] t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} + (\zeta_a -$
        $1)\mathbf{v}.^{-1} + (\zeta_b - 1)(1 - \mathbf{v}).^{-1}\}$;
    end
    for $\mathbf{a_k} \in \mathbf{a}$ do
        $\mathbf{a_k} = \mathbf{a_k} - \lambda_t[-\alpha \Sigma_{x_{u,v,k} \in \mathbf{x}} \mathbf{v} t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} - \frac{\mathbf{a_k}}{\pi\sigma_a^2}]$;
    end
end
```



(a) Basic (b) CPV (c) CPP

Figure 1: Plate notation for utility models

6. Utility Model with MNAR Observations

The "absolute" strategy in model basic is in its essence a model with missing at random observations. When the observations are not missing at random, as shown in Sec. 3, the optimal parameters should be inferred by maximizing the joint likelihood of responses and feedback. Depending on the relationship between $x_{u,v,k}$ and $r_{u,v,k}$, we present two models, i.e. conditional probability on the value of feedback (CPV) and conditional probability on the pair-wise preference (CPP).

6.1. Model CPV

In model CPV, we assume that the probability of producing a response is dependent on the value of feedback. The conditional probability $p(r|x)$ is denoted as Π, where $\forall x, \Sigma_r \pi_{r,x} = 1, \forall r, \pi_{r,x} > 0$. For example, $\pi_{1,1} > \pi_{0,1}$ indicates that users are more willing to brag about a positive purchase. Model CPV has the following generation process.

- Generate user preference from a multivariate Gaussian distribution $u \sim \mathcal{N}(u; 0, \sigma_u^2)$, contextual preference $a_k \sim \mathcal{N}(a_k; 0, \sigma_a^2)$, item specific feature strength $v_c \sim Beta(v_c; \zeta_a, \zeta_b)$ for each user, item-feature, and context respectfully. Generate global vocabulary distribution over the feature phrases $\theta \sim Dirichlet(\theta; \xi)$.
- For each review written by u on v under context k
  - Generate contextual polarity $x \sim Bern(x; g(UC_k(a, u, v)))$

  - Generate response $r \sim Bernr; \Pi_x$
  - For each product feature
    - $*$ Choose feature phrases $c \sim Discrete(c; \theta)$
    - $*$ Select an opinion polarity to describe the feature $o \sim Bern(o; v_c)$

We apply an EM algorithm for inference. In the M-step, instead of maximizing the expection of joint log-likelihood, we conduct a one-step update along the line search direction. The inference is shown in Alg. 2.

6.2. Model CPP

Empirical studies [33] discovered the phenomenan of extreme reviews. Intuitively, a stronger emotion, either positive or negative, is more likely to trigger a response. In this sense, model CPP explains the generation of extreme reviews by introducing an auxiliary $y_{u,v,k}$, which is the pairwise comparison between contextual and non contextual utility. It assumes that the response is not only related to the contextual utility, but also dependent on the non contextual utility. As shown in Tab. 3, the reviewers respond to extreme circumstances, i.e. if the user is satisfied with the product under the context $x_{u,v,k} = 1$, and the satisfaction is larger than the non contextual satisfaction $y_{u,v,k} = 1$, then the user will post a comment; if the user is disappointed with the product under the context $x_{u,v,k} = 0$, and the anger is stronger than the non contextual counterpart $y_{u,v,k} = 0$, then a comment is expected; otherwise, the comments will be missing.

As shown in Fig. 5.2, we present the following generation process.

- Generate user preference from a multivariate Gaussian distribution $u \sim \mathcal{N}(u; 0, \sigma_u^2)$, contextual preference $a_k \sim \mathcal{N}(a_k; 0, \sigma_a^2)$, item specific feature strength $v_c \sim Beta(v_c; \zeta_a, \zeta_b)$ for each user, item-feature, and context respectfully. Generate global vocabulary distribution over the feature phrases $\theta \sim Dirichlet(\theta; \xi)$.

- For each review written by u on v under context k
  - Generate contextual polarity $x \sim Bern(x; g(UC_k(a, u, v)))$
  - Generate conventional polarity $y \sim Bern(y; g(UC_k(a, u, v) - UC(u, v)))$
  - Generate response $r \sim Discrete\pi$
  - For each product feature
    - $*$ Choose feature phrases $c \sim Discrete(c; \theta)$

Algorithm 2: Learning Process for model CPV

```
Input: A set of positive and negative feedback $\mathbf{x}$, a set of feature opinion
       pairs $\{< c, o >\}$
Output: $\mathbf{a}, \mathbf{V}, \mathbf{U}, \Pi$
Initialize $\mathbf{a}, \mathbf{V}, \mathbf{U}, \Pi$;
while not convergent do
    E-Step;
    for $x_{u,v,k} \in \mathbf{x}$ do
        $q(x, u, v, k) =$
        $\Pi_{x,1}^{r_{u,v,k}}(1 - \Pi_{x,1})^{(1-r_{u,v,k})} \frac{1}{1+\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]}^{x_{u,v,k}} \frac{\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]}{1+\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]}^{1-x_{u,v,k}}$;

        Normalize $q(x, u, v, k)$ over $x \in \{0, 1\}$;
    end
    M-step;
    for $x_{u,v,k} \in \mathbf{x}$ do
        $t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} = \frac{\{-\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]\}^{x_{u,v,k}} 1^{1-x_{u,v,k}}}{1+\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]}$;
    end
    for $\mathbf{u} \in \mathbf{U}$ do
        $\mathbf{u} = \mathbf{u} - \lambda_t[-(1-\alpha)\Sigma_k\Sigma_{x_{u,v,k}\in\mathbf{x}} t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}}\mathbf{v} - (1 -$
        $\alpha)\Sigma_k\Sigma_{x_{u,v,k}\in\mathbf{x}^{mis}}\Sigma_x q(x, u, v, k) t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}}\mathbf{v} - \frac{\mathbf{u}}{\pi\sigma_u^2}]$;
    end
    for $\mathbf{v} \in \mathbf{V}$ do
        $\mathbf{v} = \mathbf{v} - \lambda_t\{\Sigma_k\Sigma_{x_{u,v,k}\in\mathbf{x}}[-\alpha\mathbf{a_k} - (1-\alpha)\mathbf{u}]t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} +$
        $\Sigma_k\Sigma_{x_{u,v,k}\in\mathbf{x}^{mis}}[-\alpha\mathbf{a_k} - (1-\alpha)\mathbf{u}]\Sigma_x q(x, u, v, k) t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} +$
        $(\zeta_a - 1)\mathbf{v}.^{-1} + (\zeta_b - 1)(1 - \mathbf{v}).^{-1}\}$;
    end
    for $\mathbf{a_k} \in \mathbf{a}$ do
        $\mathbf{a_k} = \mathbf{a_k} - \lambda_t[-\alpha\Sigma_{x_{u,v,k}\in\mathbf{x}}\mathbf{v} t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} -$
        $\alpha\Sigma_{x_{u,v,k}\in\mathbf{x}^{mis}}\mathbf{v}\Sigma_x q(x, u, v, k) t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} - \frac{\mathbf{a_k}}{\pi\sigma_a^2}]$;
    end
    for $x \in \{0, 1\}$ do
        $\Pi_{x,1} = \frac{\Sigma_{x_{u,v,k}\in\mathbf{x}} r_{u,v,k} + \Sigma_{x_{u,v,k}\in\mathbf{x}^{mis}} q(x,u,v,k) r_{u,v,k} + \zeta_a - 1}{|\mathbf{x}| + Sigma_{x_{u,v,k}\in\mathbf{x}^{mis}} q(x,u,v,k) + \Sigma j \in a,b(\zeta_j - 1)}$;
    end
end
```

Table 3: The NOR distribution of $p(r|x, y)$

$p(r\|x, y)$	x	y
1	1	1
1	0	0
0	1	0
0	0	1

$*$ Select an opinion polarity to describe the feature $o \sim Bern(o; v_c)$

To maximize the likelihood in Equ. 8, we present the EM algorithm in Alg. 3. Similar to Alg. 2, we increase the objective by line search update in the M-step.

$$\begin{aligned} p(\mathbf{x}, \mathbf{r}, \mathbf{c}, \mathbf{U}, \mathbf{V}, \mathbf{a}|\mathbf{c}, \sigma_u, \sigma_a, \zeta_a, \zeta_b, \xi) &= \\ &\Pi_{x_{u,v,k} \in \mathbf{x}}[p(x_{u,v,k}|\mathbf{u}, \mathbf{v}, \mathbf{a}_{\mathbf{k}})p(y_{u,v,k} = x_{u,v,k}|\mathbf{u}, \mathbf{v}, \mathbf{a}_{\mathbf{k}})] \\ &\Pi_{x_{u,v,k} \in \mathbf{x}^{mis}}[\Sigma_x p(x_{u,v,k} = x|\mathbf{u}, \mathbf{v}, \mathbf{a}_{\mathbf{k}})p(y_{u,v,k} \neq x|\mathbf{u}, \mathbf{v}, \mathbf{a}_{\mathbf{k}})] \\ &p(\mathbf{U}|\sigma_u)p(\mathbf{V}|\zeta_a, \zeta_b)p(\mathbf{a}|sigma_a)\Pi_{o_{c,v}}p(o_{c,v}|c, v) \end{aligned} \tag{8}$$

7. Experiment

In this section, we want to study the following questions. (1) How do we evaluate review guided context aware recommender systems? (2) How does each component of our framework perform, compared with state-of-the-art methods? Towards this end, we design a set of experimental protocals.

7.1. Experimental Setup

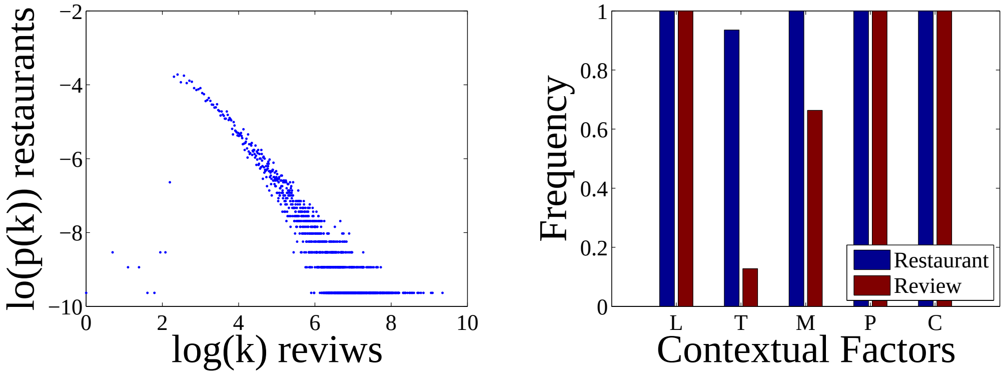
We use a real data set, which is crawled from the largest Chinese review sites[2]. We will refer this data set as the dianping data set. We crawl a total number of 2593268 reviews on 15292 restaurants located in Beijing. As shown in figure 2(a), the review distribution is a truncated power-law distribution. The log probability $logp(k)$ of restaurants with k reviews is linear with $log(k)$.

The commodity features are extracted as follows. We first manually build a dictionary containing all the possible aspects of a restaurant, example aspects

[2] dianping.com

Algorithm 3: Learning Process for model CPP

```
Input: A set of positive and negative feedback $\mathbf{x}$, a set of implicit feedback $\mathbf{x}^{mis}$, a set of feature opinion pairs $\{< c, o >\}$
Output: $\mathbf{a}, \mathbf{V}, \mathbf{U}$
Initialize $\mathbf{a}, \mathbf{V}, \mathbf{U}$;
while not convergent do
    E-step: for $r_{u,v,k} = 0$ do
        for $x, y \in \{0, 1\}$ do
            $q(x_{u,v,k}, y_{u,v,k}) = \frac{1^{x_{u,v,k}} exp[(-\alpha\mathbf{u} - (1-\alpha)\mathbf{a_k})\mathbf{v}]^{(1-x_{u,v,k})}}{1+exp[(-\alpha\mathbf{u}-(1-\alpha)\mathbf{a_k})\mathbf{v}]} \frac{1^{y_{u,v,k}} exp(-\alpha\mathbf{a_k}+\alpha\mathbf{u})\mathbf{v}^{(1-y_{u,v,k})}}{1+exp[(-\alpha\mathbf{a_k}+\alpha\mathbf{u})\mathbf{v}]} p(r_{u,v,k}|x_{u,v,k}, y_{u,v,k})$;
        end
        Normalize $q(x_{u,v,k}, y_{u,v,k}|r_{u,v,k}, \mathbf{v}, \mathbf{u}, \mathbf{a_k})$;
    end
    M-step: for $x_{u,v,k} \in \mathbf{x}$ do
        $t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} = \frac{\{-\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]\}^{x_{u,v,k}} 1^{1-x_{u,v,k}}}{1+\exp[(-\alpha\mathbf{a_k}-(1-\alpha)\mathbf{u})\mathbf{v}]}$;
        $s_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} = \frac{\{-\exp[(-\alpha\mathbf{a_k}+\alpha\mathbf{u})\mathbf{v}]\}^{y_{u,v,k}} 1^{1-y_{u,v,k}}}{1+\exp[(-\alpha\mathbf{a_k}+\alpha\mathbf{u})\mathbf{v}]}$;
    end
    for $\mathbf{u} \in \mathbf{U}$ do
        $\mathbf{u} = \mathbf{u} - \lambda_t[\Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}}(-(1-\alpha)t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} + \alpha s_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}})\mathbf{v} + \Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}^{mis}} \Sigma_{x,y \in \{0,1\}}(-(1-\alpha)q(x,y)q(x_{u,v,k}, y_{u,v,k})t_{x_{u,v,k}=x,\mathbf{a_k},\mathbf{v},\mathbf{u}} + \alpha s_{x_{u,v,k}=1-x,\mathbf{a_k},\mathbf{v},\mathbf{u}})\mathbf{v} - \frac{\mathbf{u}}{\pi\sigma_u^2}]$;
    end
    for $\mathbf{v} \in \mathbf{V}$ do
        $\mathbf{v} = \mathbf{v} - \lambda_t\{\Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}}[(-\alpha\mathbf{a_k} - (1-\alpha)\mathbf{u})t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} + (-\alpha\mathbf{a_k} + \alpha\mathbf{u})s_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}}] + \Sigma_k \Sigma_{x_{u,v,k} \in \mathbf{x}^{mis}} \Sigma_{x,y \in \{0,1\}} q(x_{u,v,k}, y_{u,v,k})[(-\alpha\mathbf{a_k} - (1-\alpha)\mathbf{u})t_{x_{u,v,k}=x,\mathbf{a_k},\mathbf{v},\mathbf{u}} + (-\alpha\mathbf{a_k} + \alpha\mathbf{u})s_{x_{u,v,k}=1-x,\mathbf{a_k},\mathbf{v},\mathbf{u}}] + (\zeta_a - 1)\mathbf{v}.^{-1} + (\zeta_b - 1)(1 - \mathbf{v}).^{-1}\}$;
    end
    for $\mathbf{a_k} \in \mathbf{a}$ do
        $\mathbf{a_k} = \mathbf{a_k} - \lambda_t[-\alpha\Sigma_{x_{u,v,k} \in \mathbf{x}}[\mathbf{v}t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} - \alpha s_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}}] + -\alpha\Sigma_{x_{u,v,k} \in \mathbf{x}^{mis}}[\Sigma_{x,y \in \{0,1\}} \mathbf{v}t_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}} - \alpha s_{x_{u,v,k},\mathbf{a_k},\mathbf{v},\mathbf{u}}]q(x_{u,v,k}, y_{u,v,k}) - \frac{\mathbf{a_k}}{\pi\sigma_a^2}]$;
    end
end
```



(a) Distribution of reviews

(b) Distribution of features

Figure 2: Statistics of data sets

include “food”,“parking”,“service” and so on. Then we run a sentiment classifier [18] to extract the opinion (positive or negative) of the review on each aspect. Thus the commodity features are pairs consisting of an aspect and an opinion polarity, e.g. “food+” is an extracted feature for the following review phrase “very good food”. In the dianping data set, the price is sometimes demonstrated with the review. In case the price is missing in the review, we use the average price of the restaurant.

The predefined categories for contextual factors used in our experiments include time,location,companion,price and cuisine. We define 5 time factors, including morning, noon, afternoon, night and weekend. Time phrases in a review are matched to one of the 5 time factors. The location factors are collected from dianping, including 43 commercial districts in Beijing. The location factors are identical to all reviews corresponding to the same restaurant. There are 4 types of companion factors(boss, friend, family, couple) and 5 different ranges of number of companions($1, 2, 3-4, 5-10, \geq 10$).The price factors are preprocessed to be 7 categories($20, 20-50, 50-80, 80-120, 120-200, 200-500, 500-800, \geq 800$). The cuisine factors contain 95 types of cuisines.

All commodity features and contextual factors are binary valued. We count the probability of restaurants and reviews that the contextual factors appear for more than once. As shown in figure 2(b), cuisine, location and price factors (denoted as “C”,“L”,and “P” in the axis) appear in all restaurants and reviews. A fairy number of people take notes on who they eat with(companion factors denoted as “M” in the axis), almost all restaurants have at least one record with companion

Table 4: The top contextual factors in each group with the largest b_k^x

Context	Business	Party	Family	Dating	Leisure
Time	noon	afternoon	noon	weekend	weekend
Loc.	E.40th St.	Houhai	Zoo	Sanlitun	Houhai
Comp.	boss	friend	family	family	family
No. Comp.	5-10	3-4	3-4	2	3-4
Price	200-500	50-80	200-500	80-120	120-200
Cuisine	roast duck	hotpot	Russian	ryori	fastfood

information. Fewer reviews mention when (time factors denoted as “T” in the axis) the experience happens, but the percentage of restaurants with at least one review containing time phrases is much larger.

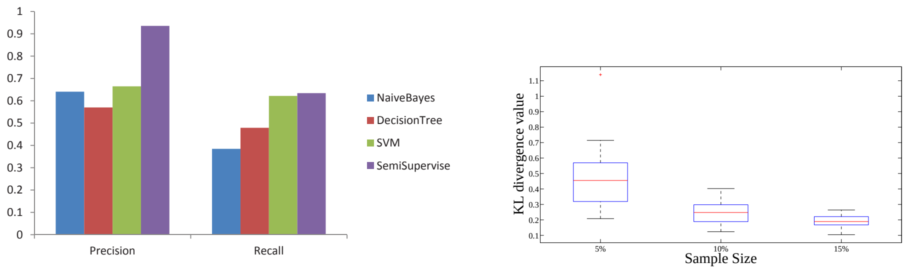
We define 5 contexts, namely “Business Banquet”, “Party”, “Family Get-together”,“ Dating” and “ Eat for Leisure”. We pick 27 most popular restaurants and annotate randomly 15% of the reviews (17662 reviews). We observe that the context distribution is unbalanced. Most reviews (61%) are labeled as irrelevant to any context, which suggests the predominance of implicit feedback. 19% reviews are labeled to be positive about “Party”. We sample 25% of the size of positive instances from the collection of reviews related to each contexts $\bar{x}$, and construct a relatively balanced negative training set $E^{-}x$, with approximately 125% of the size of $E^{+}x$. In annotation and opinion categorization, we only deal with reviews with positive opinions. That is, we first filter out reviews with negative ratings, for a “cleaner” training set.

The step size is set to be $\alpha = 1$. The coefficient related to variable std, denoted by $\lambda_p, \lambda_u, \lambda_c$ are decided by cross-validation. For the results below, the values are set to be $\lambda_p = \frac{1}{|E|}, \lambda_u = 1, \lambda_c = 0.01$. The combination weight for contextual and user preference is set to be $\alpha = 0.5$.

7.2. Context Categorization Performance

We first analyze the capability of the context classification model of selecting the most sensitive factors to each context. Table 7.2 shows the top contextual factors selected with largest $b_{k,f}$ in time, location, companion, number of companions, price and cuisine factors for each context k. Consistent with our common sense, people tend to pay more for business banquet and less for a party, arrange datingin weekends, and get together with friends in a party.

We next analyze how the semi-supervised mechanism improves the categorization. The comparative methods are Naive Bayes, SVM, and Decision Tree.



(a) Comparative categorization performance

(b) D_{KL} with different sample sizes

Figure 3: Performance of contextual categorization

Table 5: Comparative performance of cold-start recommendation

Context	dating	business	family	party	leisure
MostPop	0.7	0.2	0.1	0.8	0.1
TextRating	0.6	0.1	**0.2**	0.4	0.4
CLDA	**0.8**	0.4	0.1	0.8	0.1
Basic-absolute	**0.8**	**0.7**	**0.2**	**1.0**	**1.0**

The evaluation metrics are precision and recall rates averaged across all contexts. As shown in Fig. 3(a), traditional classifiers suffer from the insufficiency of training instances. The semi-supervision obtains best performance in both metrics. The improvement of precision is in particular significant. Due to the fact of "silent majority", a constraint in prior distribution leads to more accurate classification hyperplane. Figure 3(b) shows the KL-divergence of the predicted context distribution and the prior distribution from the annotated reviews of sample size $5\%, 10\%, 15\%$ respectfully. We can see that the KL-divergence is quite small, and is decreasing as the sample size increases.

7.3. Cold-start Recommendation

We next study cold-start recommendation on which most RGCARS focus. We use the intelligent ranking provided by dianping as the gold standard. Precision At Top Ten (P@10) is the evaluation metric. The comparative methods include (1) MostPop, a naive baseline where restaurants are ordered by the number of reviews; (2)TextRating [15], a scoring system based on online review mining; (3)CLDA [11], a LDA alike model for query driven context aware recommendation. We set the query keywords to be the context; (4) our basic model with the "absolute" strategy, where $\alpha = 1$.

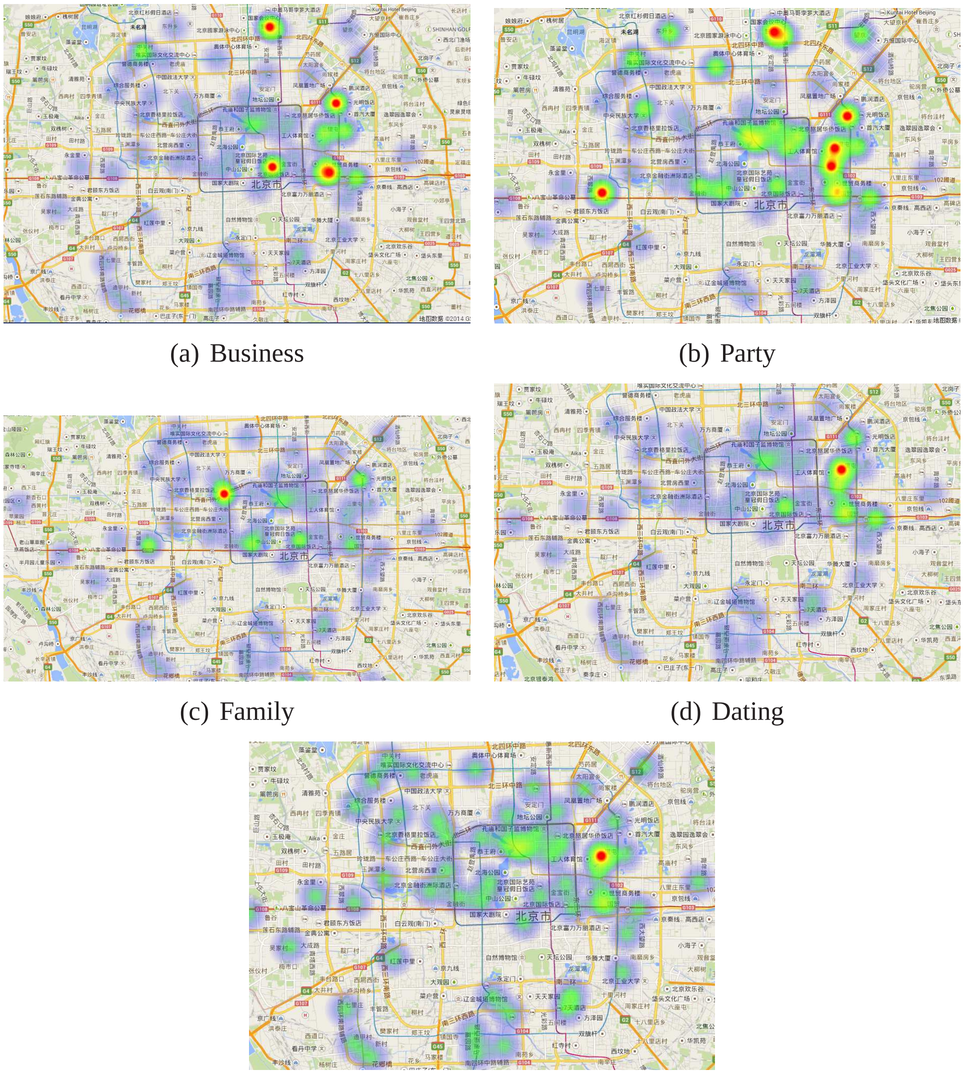
As shown in Tab. 7.3, our model achieves best results in all contexts, with all pre-procession of textual contents being identical. It shows that the utility surplus model can capture the underlying relationships among user preferences, product features, purchases and reviews. We also notice that utilizing the information in online reviews does not gurantee improvement of the cold-start recommender, especially for context “business banquet”, “dating” and “party”. This may be caused by the ambiguity of natural language. Erros in extracting the opinions and features from online reviews may cause damage to the recommender. In our model, ratings are important factors for judging the polarity of feedback. Thus our model is more false tolerant to a noisy NLP miner.

7.4. Visualization of Context-aware Landmarks

We pick the top 1000 restaurants computed by basic model in each context. The density of restaurant distribution is drawn at a map of Beijing county. As shown in figure 4(a) to figure 4(e), we have the following observations: (1) the best place for business banquet is right in the city center of Beijing; (2) best places for party are more scattered settled; (3) family can get together at any place, but the famous place of interest Xizhimen is the best choice; (4)dating and leisure eating can also take place at anywhere, but the most popular choice is quite different with that people will make for uniting with family members. The most popular choice Sunlitun is a place full of bars. The visualization accord with our knowledge about the city.

7.5. Context aware recommendation

The final task to be evaluated is context aware recommendation. We filter our users with at least 5 reviews for training, and select 10% of the reviews by users with at least 20 reviews for testing. An ideal masurement, for recommendation systems with implicit feedback, is based on predictions of all items for an user, whether they are observed or missing. When such test data is not available, rating based measurements will be imprecise. We use MRR and NDCG as the approximate evaluation metrics. MRR is the mean reciprocal rank of the highest rated restarant of the ground truth in the test set. It is targeted to acquire the best option. NDCG measures quality of the predicted rankings of all ground truth items. We compare our basic models with two strategies (basic-absolute, basic-fuzzy), two variant models (CPV and CPP) with a wide range of state-of-the-art methods, including traditional recommendation models (ItemKNN), models with implicit feedback (CLiMF [28], BPR [26]) and models based on online review mining(PLRM [17]). The results are plot in Fig. 5(a) and Fig. 5(b).



(a) Business

(b) Party

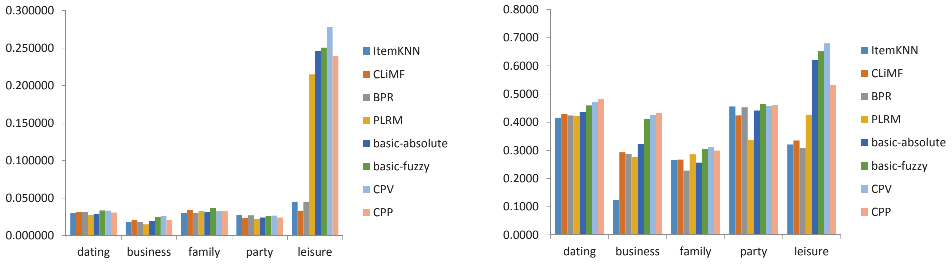
(c) Family

(d) Dating

(e) Leisure

Figure 4: Visualization on top 1000 restaurants for each contexts at the map

We have the following observations. (1) Our models generally outperform other state-of-the-art methods in terms of MRR and NDCG, for most contexts. Similar to the cold-start recommendation in Tab. 7.3, our models are significantly better for context "leisure". (2) A universal trend exists that models utilizing implicit feedback yield higher results than models which don't, i.e. CliMF and BPR are better than ItemKNN, basic-fuzzy and CPV are better than basic-absolute. This phenomenan highlights the importance of implicit feedback. (3) Compared with CPV, CPP is more sensetive to contexts. As CPP place strict constraints on extreme reviews, it might be more appropriate for less common contexts.



(a) MRR

(b) NDCG

Figure 5: Comparative performance of context aware recommender system

8. Conclusion

In this paper, we mainly focus on exploring the potencial of implicit feedback in a review guided context aware recommender system. We present new models, based on the utility surplus theory, to tackle the implicit feedback problem by treating them as complete observations or missing not at random observations. We systematically compare the assumptions and performances of thes models. The most important academic contribution of this paper is that, to the best of our knowledge, the unique types of implicit feedback (both context free and context aware) have not yet been studied by the community. Therefore our research might shed some insight into mining online reviews for recommender system, and other applications. Further research issues include adapting and testfying more assumption in the missing not at random model. For example, the inter homogenity of online communities, and applying the CPP model to appropriate contexts.

9. Acknowledgement

Chen Lin is partially supported by China Natural Science Foundation under Grant Nos. NSFC61102136, NSFC61472335, CCF-Tencent Open Research Fund under Grant No. CCF-Tencent20130101, Base Research Project of Shenzhen Bureau of Science,Technology, and Information under Grand No. JCYJ20120618155655087, Baidu Open Research under Grant No.Z153283.

References

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

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[3] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context-aware places of interest recommendations for mobile users. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, pages 531–540. Springer, 2011.

[4] Linas Baltrunas and Francesco Ricci. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 245–248, New York, NY, USA, 2009. ACM.

[5] R.M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 43–52. Ieee, 2007.

[6] Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. An approach to social recommendation for context-aware mobile services. *ACM Trans. Intell. Syst. Technol.*, 4(1):10:1–10:31, February 2013.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez. Recommender systems survey. *Knowledge-Based Systems*, 46(0):109 – 132, 2013.

[8] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, and Wei-Ying Ma. Musicsense: Contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 553–556, New York, NY, USA, 2007. ACM.

[9] Michele Gorgoglione, Umberto Panniello, and Alexander Tuzhilin. The effect of context-aware recommendations on customer purchasing behavior and trust. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 85–92, New York, NY, USA, 2011. ACM.

[10] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 131–138, New York, NY, USA, 2012. ACM.

[11] Negar Hariri, Bamshad Mobasher, and Robin Burke. Query-driven context aware recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 9–16, New York, NY, USA, 2013. ACM.

[12] David Heckerman and Christopher Meek. Models and selection criteria for regression and classification. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, pages 223–228, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[13] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 79–86, New York, NY, USA, 2010. ACM.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[15] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 115–122, New York, NY, USA, 2012. ACM.

[16] Beibei Li, Anindya Ghose, and Panagiotis G. Ipeirotis. Towards a theory model for product search. In *Proceedings of the 20th international conference on world wide web*, pages 327–336, 2011.

[17] Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. Contextual recommendation based on text mining. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 692–700, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[18] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA, 2005. ACM.

[19] Hongyan Liu, Jun He, Tingting Wang, Wenting Song, and Xiaoyang Du. Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*, 12(1):14 – 23, 2013.

[20] Samaneh Moghaddam and Martin Ester. The FLDA model for aspect-based opinion mining: addressing the cold start problem. In *Proceedings of the 22nd international conference on World Wide Web*, pages 909–918. International World Wide Web Conferences Steering Committee, May 2013.

[21] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[22] C. Palmisano, A Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on*, 20(11):1535–1549, Nov 2008.

[23] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 265–268, New York, NY, USA, 2009. ACM.

[24] Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 913–921, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[25] Sindhu Raghavan, Suriya Gunasekar, and Joydeep Ghosh. Review quality aware collaborative filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 123–130, New York, NY, USA, 2012. ACM.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[27] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.

[28] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, New York, NY, USA, 2012. ACM.

[29] Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 269–272, New York, NY, USA, 2010. ACM.

[30] Jianfeng Si, Qing Li, Tieyun Qian, and Xiaotie Deng. Users interest grouping from online reviews based on topic frequency and order. In *World Wide Web*, 17(6):1321C1342, 2014.

[31] S. Wang, B. Zou, C. Li, K. Zhao, Q. Liu, and H. Chen. Crown: A context-aware recommender for web news. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1420–1423, April 2015.

[32] Marlin, Benjamin M. and Zemel, Richard S. Collaborative Prediction and Ranking with Non-random Missing Data In *Proceedings of the Third ACM*

Conference on Recommender Systems, RecSys ’09, pages 5–12, New York, NY, USA, 2009, ACM.

[33] A.C. Wojnicki, D. Godes. Word-of-Mouth as Self-Enhancement In *HBS Marketing Research Paper No. 06-01*, 2008.