Review Guided Context Aware Restaurant Recommendation

paper ID:70

ABSTRACT

Context aware recommendation suffers from data sparsity problem. In this paper, we propose a principled two-level context aware recommendation framework based on online reviews for restaurant recommendation. In the first level, the implicit contexts are inferred from user inputs through a multi-class classifier, which is learnt from nosily labeled and unlabeled online reviews in a semi-supervised manner. We present a sampling scheme to generate accurate and balanced training sets. In the second level, the restaurants are ranked according to their utility surplus, parameterized by context aware user preferences over restaurant features. The hidden restaurant features are also learnt from opinionated online reviews. Experiments on real data sets demonstrate the effectiveness of the proposed framework.

1. INTRODUCTION

Recent years have witnessed the emergence of recommendation systems. However, naive recommenders ignore the fact that purchases of products and services are usually redeemed within certain "context". A context is a set of conditions, including physical, social and even psychological conditions, under which a consumer is expected to make consumption. We will use the following example to illustrate that a context independent mechanism will limit the prediction power of a naive recommender.

Example: Alice is looking for a place where she can have her weekday lunch alone. A sandwich combo seems to be an excellent choice. The absolute price of the item, along with the cleanliness and quick service of the restaurant are dominant factors under the context "weekday lunch for one person". In the meantime, Bob is preparing for a date with his girlfriend at Christmas' Eve. This time, a coupon on some nice restaurants serving French cuisine and fine wine seems to be more preferable. A reasonably higher price is acceptable, as long as it is offered at a great discount.

In this paper, we focus on the context aware restaurant recommendation problem. For a set of contextual factors, restaurants are ranked according to how they suit the implied context. To be more specific, we gather contextual factors including when, where, with whom and other freely expressed contextual information(i.e. "with a convenient parking lot") that the desired restaurant is applied; infer the implicit context (i.e. business banquets, dating couples, etc.); quantify how the restaurants can satisfy users' needs in the specific context and output restaurants with maximal satisfaction.

Context aware recommendation systems (CARS) have been successfully applied in a few areas, including travel [3], music [4] and so on. There are generally 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 [1]), the second type is to post-filter (i.e. split items to different contexts [2]), the third type is to model the context (i.e. as a latent variable in BNN [9], or as a latent factor in matrix factorization [6]). The sparsity of rating data is an obstacle for CARS, because traditional CARS make use of explicit user feedback (i.e. ratings). Lack of user feedbacks will lead to inaccurate predictions.

As more and more people share shopping experiences in review sites, it is possible that we generate recommendations based on textual contents in online reviews, instead of explicit ratings. A few researchers begin to exploit the feasibility of learning context aware user preferences from online reviews [7]. However, two challenges remain in producing accurate predictions. Firstly, due to the high volume of online reviews, manually labeling enough context related opinions are impossible. Tags associated with each review are usually noisy. It is more practical to learn from a few roughly labeled instances and a large number of unlabeled reviews. Secondly, the exact restaurant features are partially observable through the consumption experiences expressed in the corresponding reviews. The recommendation performance will be deteriorated if inaccurate features are used in the restaurant profile.

To address the above two challenges, we integrate opinion mining and semi-supervised learning techniques in a probabilistic generative model. The generative model consists of two levels. In the first level, it infers the implied context from user input contextual factors. In the second level, it aggregates the probabilities of choosing a restaurant in each context. We present a formal preference quantification model based on utility surplus to compute the probability of choosing a restaurant in a particular context. The utility surplus is modeled as a linear combination of context aware user preferences over the hidden restaurant price and features, which are learnt from opinionated online reviews. We present a semi-supervised multinomial logistic regression model to classify the opinions expressed in online reviews. In order to overcome the noisy tagging and class imbalance problems, we present a sampling scheme to generate precise and balanced training sets.

~~2. METHODOLOGY~~

2.1 Recommendation Model

Given a query q, consisting of some contextual factors(i.e. "with girlfriend, at Christmas Eve."), the context aware recommender aims to rank restaurants according to how they satisfy user needs in the implied context. We assign a probabilistic score to each restaurant, denoted as $p(c|q)$, the probability of restaurant c being chosen for query q. Note that although we require an input query q, the methodology can be easily expanded to the usual multi-dimensional $User \times Item \times Context$ settings.

Since the query may match several contexts (i.e. "with girlfriend, at Christmas Eve." can implies "dating", "party" etc), the probabil-

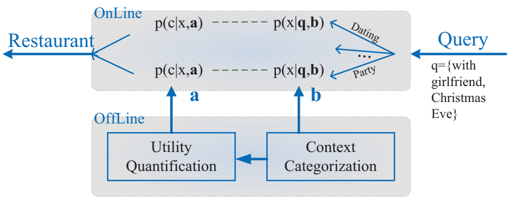

Figure 1: Framework for context aware recommendation

ity can be rewritten as $p(c|q) = \Sigma_x p(c|x)p(x|q)$, where x is a context, the first term is the utility of each restaurant under context x, and the second term is the probability of categorizing query q to context x.

Consequently, the recommendation framework depicted in Figure 1 consists of an off-line learning process and an online recommendation process. The off-line learning process consists of two levels. In the first level, a context classifier parameterized by $\mathbf{b}$ is trained to predict context label for each unlabeled review e. In the second level, the utility quantification model parameterized by $\mathbf{a}$ is learnt for each context from labeled and predicted reviews. The online recommendation uses both $\mathbf{a}$, $\mathbf{b}$ to estimate the final possibility.

2.2 Context Categorization

Online reviews for restaurants can be of different formats. We consider the most common type: each review is a brief text message with multiple tags. Suppose we predefine totally K contexts and the corresponding tags, then the tagged reviews can be grouped into $K+1$ categories: $\{E^+(x_1), \cdots, E^+(x_K), E^+(x_{K+1})\}$, where $E^+(x_1), \cdots, E^+(x_K)$ are reviews tagged as context $x_1, \cdots, x_K$ respectfully. The last category consists of reviews that do not belong to any context. The objective of context categorization is to infer the probability $p(x|\mathbf{f})$ of review f being context x for a contextual factor vector $\mathbf{f}$. A multinomial logistic regression for this multi-class categorization problem is to minimize:

$$\min_{\mathbf{b}} -\sum_{k,f \in E^+(x_k)} I(l(f)=k) \log \frac{\exp \mathbf{b^k f}}{\Sigma_k \exp \mathbf{b^k f}}, \tag{1}$$

where $I(l(f) = k)$ is an indicator function that the review is labeled as a tag of context x^k by online users. The context distribution of $\mathbf{f}$ is $p(l(f) = k|\mathbf{f}) = \frac{\exp \mathbf{b}^k \mathbf{f}^T}{\Sigma_{k'} \exp \mathbf{b}^{k'} \mathbf{f}^T}$.

A critical issue is that the tags are not always accurate. A consumer may express opinions about the consumption experience in the text message, but select tags that do not correspond to the current experience. For example, a review "Nice place for a night date" are tagged as "party" and "dating". Our basic assumption is that each time a consumption is completed under one particular context, which suggests that each review is associated with exactly one label. However, when tagging the review, the reviewer may combine the "true" label with labels from other reviews, resulting multi-labeled reviews. Noisy tagging will degrade the classification performance. We need to train the context classifier with the "true" label for reviews with multiple labels.

Suppose a review e is associated with M labels, then the probability of choosing a label l is $p_e(l) = \frac{1}{M}, l \in L(e)$. Let the probability distribution for labels in the complete set of reviews on the given restaurant be p_b, then the probability of choosing the "true" label for e is $p(l(e) = l) = \alpha p_b(l) + (1-\alpha) p_e(l)$. Intuitively, when the size of observed reviews is large, then the background probability distribution p_b is more reliable. We set $\alpha = \frac{1}{(1+e^{-\beta \times N})}$, where N is the number of reviews on the given restaurant. We use the above assumption to generate training and test set. In generating the training set for context x_k, we sample the positive instance for each review e with probability $p(l(e) \in x_k) = \Sigma_{l \in x_k} p(l(e) = l)$. The test set contains unlabeled reviews where the overall sentimental polarity is positive (i.e. the reviews express positive opinions on unknown contexts).

Another challenging issue is that the number of tagged reviews are relatively small. However, there are hidden structures among online reviews, which can be very useful in predicting contexts. We have the following assumptions. (1) All reviews related to the same restaurant share a prior distribution of contexts. We can use the prior to guide the context prediction for unlabeled reviews. For example, if the restaurant has a context distribution $P\tilde{X}^c = \{$"party:"0.1,"banquet:"0.8,"weekday lunch:"0.1$\}$ in the sample set, then we expect the context "banquet" is dominant in the unlabeled reviews. (2) All reviews are written by the same group of people, which means that, the prior distribution of context and non-context related reviews, denoted as $\tilde{PU}$, are shared among all reviews. For example, if in a random sample, 33% percent of reviews are related to at least one context, then we would expect that about 67% percent of unlabeled reviews are not related to any context. Note that $P\tilde{X}^c$ is a K-dimensional vector, because a restaurant is always related to at least one context, and $\tilde{PU}$ is a 2-dimensional vector, because a review can be unrelated to any context, so we construct $\tilde{P}^c$ as a $K+1$-dimensional vector:

$$\tilde{P}^c_k = \begin{cases} \tilde{PU}_1 P\tilde{X}^c_k & 1 \le k \le K; \\ \tilde{PU}_2 & k = K. \end{cases}$$

Following the aforementioned assumptions, we devise the KL-divergence between the distribution of review samples and the estimated distribution in unlabeled reviews $p^c(l|\mathbf{b})$ based on the parameter $\mathbf{b}$:

$$D(\tilde{p}\|p(l|\mathbf{b})) = \sum_c D_{KL}(\tilde{p^c}\|p^c(l|\mathbf{b})),$$

Given a set of labeled reviews E^+ on restaurants $C^+ = \{c\}$, and a large number of unlabeled reviews EU, in order to learn the classifier parameters $\mathbf{b}$, we first compute the prior distribution $\tilde{p^c}$, and then optimize the following context categorization model (C-CM):

$$\min_{\mathbf{b}} -\frac{1}{E^+} \log p(E^+|\mathbf{b}) + \lambda_p D(\tilde{p}\|p(l|\mathbf{b})) + \lambda_c \|\mathbf{b}\|_2, \tag{2}$$

where the first term is to maximize the likelihood for generating the labeled reviews, $\frac{1}{E^+} \log p(E^+|\mathbf{b}) = \frac{1}{E^+} \sum_{f \in E^+(k)} \log p(l(f) = k|\mathbf{b})$, for all reviews labeled as context x_k; the second term is to maximize the similarity between context distribution from sampled annotations and unlabeled data; and the third term is a regularizer to avoid over-fitting.

2.3 Utility Quantification

Utility is an important property of any commodity. It measures the satisfaction people get by purchasing a commodity. Money, as a special case of commodity, can also be measured by a utility function. Intuitively, if a review is positive about the consumption experience under context x, then the context dependent utility of the restaurant is larger than the loss of utility by paying the price. In another word, the utility surplus $US^x(c) = UC^x(c) - UM^x(c_p)$, which is the context dependent utility of restaurant c minus the money utility of price c_p, will be positive.

Usually, the commodity utility can be modeled as a linear combination of user preferences over commodity features $UC(c) = \sum_k a_k c_k$, where a_k is the preference to the kth feature c_k. 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. $UM(c_P) = a_P(-c_P)$. Here a_P can also be interpreted as user preference to price. This leads to the following utility function.

$$US^x(c) = UC^x(c) - UM^x(c_P) = \sum_k a_k^x c_k + a_P^x c_P. \quad (3)$$

We make the following manipulation to ensure the score for each restaurant coincides with a probability. For a given context x, the probability of c being chosen is $p(c|x)$. The utility quantification model (UQM) is:

$$p(c|x) = g(US^x(c)) = \frac{1}{1 + \exp -\mathbf{a^x}\mathbf{c}^T}, \quad (4)$$

where $\mathbf{a^x}$ is a vector indicating user preferences over commodity features and price, as well as bias, $\mathbf{a^x} = \{a_0^x, a_1^x, \cdots, a_K^x, a_p^x\}$, and $\mathbf{c}$ is the restaurant features and price, as well as a constant feature for bias, $\mathbf{c} = \{1, c_1, \cdots, c_K, c_P\}$. If $US^x(c) > 0$, then $p(c|x) > 0.5$, which suggests that it is possible to choose the restaurant c for context x.

The user preferences $\mathbf{a^x}$ are hidden and can only be learnt from a set of labeled reviews. Moreover, the restaurant feature vector $\mathbf{c}$ is in general hidden and can only be partially observed in the reviews. On one hand, due to the capacity limit of a review, a consumer will reveal only a small portion of restaurant features. On the other hand, the individual consumption experience may differ due to unpredictable reasons. For example, it is possible that "restaurant A" is known for its world-class service, yet some consumers may encounter a harsh waiter. For the above reasons, we utilize opinion mining techniques to infer the restaurant features $\mathbf{c}$. We define $\mathbf{c}$ as a numeric vector on K aspects, where each element $c_k \in (0, 1)$ indicates the probability of a consumer experiencing a positive experience on the aspect. For example, $c_k = 0.9$ for the kth aspect "service" suggests that the restaurant consistently deliver high quality service. We use $\mathbf{e}$ to denote the features observed in review e, which is an combination of the aspect and the opinion polarity. We assume the following generating process. In order to write a review, a consumer first randomly chooses $|\mathbf{e}|$ aspects from K aspects based on a uniform distribution, then chooses the opinion polarity from a Burnoulli distribution(i.e. the probability of observing positive opinion on the given aspect $p(\text{sign}(e_k) = +) = c_k$).

Our basic idea is to use the reviews as positive and negative training samples to learn $\mathbf{a^x}$ and $\mathbf{c}$. We denote the collection of positive training instances in context x as $L^+(x)_c = \{< \mathbf{e}, l(e)_x >\}$. For a labeled review $l(e)_x = 1$. For predicted instances, $l(e)_x = p(l(e) = x|\mathbf{b})$ by Section 2.2. Similarly, we generate the collection of negative training samples $L^-(x)_c = \{< \mathbf{e}, l(e)_x >\}, l(e)_x = 0$, including reviews which are not labeled by any tag in context x, and predicted negative reviews. We have to minimize the following regularized log-likelihood for each context x:

$$\min_{\mathbf{a}^x, \mathbf{c}} -\Sigma_{\mathbf{e}} \log p(l(e)_x, \mathbf{e}) = -\Sigma_{\mathbf{e}} \log p(l(e)_x, \mathbf{e}) + \lambda_u \|\mathbf{a}^x\|_2 \quad (5)$$

where $l(e)_x \equiv 1$ for a labeled review by human annotation, and otherwise it is automatically computed by the model introduced in Section 2.2.

$$\log p(l(e)_x, \mathbf{e}) = \log p(l(e)_x|\mathbf{a}^x, \mathbf{c}) + \log p(\mathbf{e}|\mathbf{c})$$

where $\log p(\mathbf{e}|\mathbf{c}) = \Sigma_i c_i^{\text{sign}(e_i)}(1 - c_i)^{1-\text{sign}(e_i)}$, $\forall \mathbf{e} \in L^+(x)_c$, $\log p(l(e)_x|\mathbf{a}^x, \mathbf{c}) = \log \frac{1}{1+\exp(-l(e)_x \mathbf{a}^x \mathbf{c})}$, otherwise $\forall \mathbf{e} \in L^-(x)_c$, $\log p(l(e)_x|\mathbf{a}^x, \mathbf{c}) = \log \frac{\exp(-\mathbf{a}^x \mathbf{c})}{1+\exp(-\mathbf{a}^x \mathbf{c})}$

3. EXPERIMENT

The real data set used in the experiments is crawled from Chinese largest review sites dianping.com. 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. 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.com, 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. There are totally 316 contextual factors, all of which are binary valued.

The restaurant aspects are extracted from a domain independent dictionary (HIT-CIR Tongyici Cilin) containing 54 possible aspects of a restaurant, example aspects include "cuisine", "food", "parking", "service" and so on. We run a sentiment classifier [8] to extract the opinion (positive or negative) and related aspects of the review. The review features $\mathbf{e}$ are binary valued. In case the price is missing in the review, we use the average price of the restaurant.

The step size is set to be $\alpha = 1$. The value of $\lambda_p, \lambda_u, \lambda_c$ are decided by cross-validation. For the results below, the values are set to be $\lambda_p = \frac{1}{|EU|}, \lambda_u = 1, \lambda_c = 0.01$.

We first analyze the capability of the context categorization model. We define 5 contexts, namely "Business Banquet", "Party", "Family Get-together"," Dating" and " Eat for Leisure". Table 1 shows the top contextual factors selected with largest b_k^x in time, location, companion, number of companions, price and cuisine factors for each context. We can see that the right companions and the appropriate number of companions are recognized for each context. We also have some interesting discoveries. For example, people are willing to pay more for banquet and dating, people think hotpot is the best cuisine for party, etc.

Table 1: Top contextual factors with largest b_k^x

Factor	Banquet	Party	Family	Dating	Leisure
Time	noon	afternoon	noon	weekend	weekend
Loc.	E.40th St.	Houhai	Zoo	XiZhimen	ZhongGuanCun
Comp.	boss	friend	family	couple	family
No. Comp.	>10	3-4	3-4	2	3-4
Price	200-500	50-80	50-80	120-200	20-50
Cuisine	Peking	hotpot	Russian	steak	fastfood

We then pick 27 most popular restaurants and mannually annotate randomly 15% of the reviews (17662 reviews). We compare the context categorization model (CCM) with several existing methods. The features used are opinion unigrams [7], non opinionated contextual factors as in CCM, and contextual factors with word2vec expansions with similarity threshold 0.9. The classifiers include a full supervised softmax [5], semi-supervised CCM and Weka-liblinear [1]. As shown in Table 2, a one-vs.-all classifier such as liblinear performs poorly on this task. A carefully chosen ontology helps avoid noisy features, thus outperforms opinion unigrams and word2vec features. We observe that with supervision, the precisions for all types of features are boosted.

Table 2: Comparative study on context categorization

Precision	Opinion	Contextual	word2vec
Softmax	58.21%	74.24%	61.64%
CCM	58.90%	76.50%	62.50%
Liblinear	51.02%	50.94%	52.30%

We further analyze the semi-supervised mechanism in context categorization model. Figure 2 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.

[1] http://www.cs.waikato.ac.nz/ml/weka/

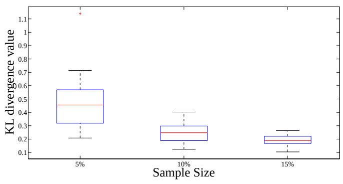

Figure 2: D_{KL} with different sample sizes

We evaluate the sampling scheme in context categorization. We show the precision for different parameter value $\beta = 1, 2 \cdots 5$ in table 3. It can be observed that the best result is obtained when $\beta = 4$. However the difference is not significant.

Table 3: ~~Parameter tunning for context categorization model~~

β	1	2	3	4	5
precision	76.50%	76.87%	77.17%	77.73%	77.10%

We next evaluate the performance of utility quantification model. Table 4 shows the top 10 restaurant features with highest a_k^x in each context. The original features are in Chinese, which are translated to English by Google translator. We have some interesting findings. For example, people are more focused on food when they are getting together with family members, while they care more about the environment when dating. Also, we found that people are more cautious to price when having a banquet and dating, this corresponds to the discovery in table 1 that people are willing to pay more under these contexts.

Table 4: Top 10 restaurant features with largest a_k^x

Banquet	Party	Family	Dating	Leisure
space+	service+	tone+	style+	tone+
preferential price+	toilet+	quantity+	layout+	atmosphere+
groupon+	private room+	traffic+	food container+	design+
service+	air+	toilet+	service+	traffic+
design+	atmosphere+	preferential price+	waiter+	waiter+
parking+	menu+	service+	parking+	match+
match+	private+	groupon+	decoration+	business hours+
style+	service+	discount+	design+	preferential price+
sofa+	design+	light+	discount+	wifi+
discount+	waiter+	business hours+	groupon+	sofa+

We compare the restaurant list of utility quantification model (UQM), with 4 other ranking baselines: (1) the default intelligent ranker in Dianping (DD), (2) the number of users(NU), (3) the number of reviews(NE), (4) the average rating(AR). The ground truth collected from dianping.com. There are 6 labels in dianping, the first 4 are equivalent to the contexts we defined, we merge the labels "eat casually" and "eat for leisure" as the context "leisure". We can see from Table 5, the precision in the top ranked restaurants $P@10$, $P@20$, $P@50$, $P@100$, $P@200$ are all very high.

Finally, we analyze the performance of context aware restaurant recommendations. We simulate a topic set of 30 queries. Each query has at least one contextual factors, and several freely expressed keywords. We then parse the contextual factors, and match the free expressed keywords to commodity features if possible. We conduct user studies on the topic set. We ask 7 undergraduate students, who have no knowledge about which recommender produces which result, to individually rate the top 10 results from the context-aware restaurant recommender (CAR) and several other comparative recommenders. The evaluation metrics include (1) preference, where 1 stands for not interested at all, and 5 stands for highly preferable; (2) interpretability, where 1 if it can not be explained by the corresponding reviews at all, and 5 if it is completely interpretable from the reviews; (3) divergence, where 1 if all the restaurants are with the same taste, and 5 if there are highly divergent options. The comparative recommenders include: (1) a naive ranker based on price; (2) a naive ranker based on date; (3)a

~~Table 5: Comparative study on restaurant ranking~~

Method	$P@10$	$P@20$	$P@30$	$P@50$	$P@100$	$P@200$
UQM	1.0	1.0	0.97	0.98	0.97	0.739
DD	0.9	0.95	0.96	0.96	0.98	0.94
NE	1.0	0.95	0.9	0.9	0.89	0.87
NU	0.7	0.6	0.7	0.68	0.69	0.77
AR	0.8	0.85	0.9	0.9	0.88	0.80

search based ranker Terrier[2], which uses the original query as input, and ranks each restaurant profile (aggregation of reviews) by BM25.

~~Table 6: User study on restaurant recommendation~~

Method	Preference	Divergence	Interpretability
Price	3.47	2.86	1.71
Date	3.47	**4.14**	1.57
Terrier	2.91	2.57	2.84
CAR	**4.03**$^{++}$	3.28	**4.16**$^{++}$

As shown in Table 6, CAR performs the best in terms of average preference and interpretability, which verifies the effectiveness and context awareness of the recommendation model. The improvements are significant, as $++$ indicates the p-value between the CAR result and the second best result is less than 1%. The naive rankers based on price and date perform poorly because they do not personalize recommendations based on the implied context. Their results are not explainable. Ranking by price tends to result in restaurants with similar style, i.e. cheap fast foods. A general purpose search engine does not perform well, because it is not customized to contextual queries, thus can not generate truly relevant results.

4. CONCLUSION

In this paper, we present a principled probabilistic framework to solve context aware restaurant recommendation. To overcome the data sparsity problem, we take advantage of the massive online reviews. We further propose the following two techniques: (1) a semi-supervised model for classifying the context of reviews; (2) a utility surplus model to estimate the value of a restaurant within a certain context, based on predicted reviews. Empirical results have demonstrated the effectiveness of the proposed models.

5. REFERENCES

[1] G. Adomavicius et.al. Incorporating contextual information in recommender systems using a multidimensional approach. TIST, 23(1):103–145.

[2] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. In RecSys '09:245–248.

[3] C. Biancalana et.al. An approach to social recommendation for context-aware mobile services. TIST 4(1):10:1–10:31.

[4] R. Cai et.al. Musicsense: Contextual music recommendation using emotional allocation modeling. In MM '07:553–556.

[5] D. Heckerman and C. Meek. Models and selection criteria for regression and classification. In UAI'97:223–228.

[6] A. Karatzoglou et.al. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In RecSys '10:79–86.

[7] Y. Li et.al. Contextual recommendation based on text mining. In COLING '10:692–700.

[8] B. Liu et.al. Opinion observer: Analyzing and comparing opinions on the web. In WWW '05:342–351.

[9] C. Palmisano et.al. Using context to improve predictive modeling of customers in personalization applications. In TKDE 20(11):1535–1549.

[2] terrier.org