

CSLIM: Contextual SLIM Recommendation Algorithms

Yong Zheng
Center for Web Intelligence
DePaul University
Chicago, Illinois, USA
yzheng8@cs.depaul.edu

Bamshad Mobasher
Center for Web Intelligence
DePaul University
Chicago, Illinois, USA
mobasher@cs.depaul.edu

Robin Burke
Center for Web Intelligence
DePaul University
Chicago, Illinois, USA
rburke@cs.depaul.edu

ABSTRACT

Context-aware recommender systems (CARS) take contextual conditions into account when providing item recommendations. In recent years, context-aware matrix factorization (CAMF) has emerged as an extension of the matrix factorization technique that also incorporates contextual conditions. In this paper, we introduce another matrix factorization approach for contextual recommendations, the contextual SLIM (CSLIM) recommendation approach. It is derived from the sparse linear method (SLIM) which was designed for Top- N recommendations in traditional recommender systems. Based on the experimental evaluations over several context-aware data sets, we demonstrate that CSLIM can be an effective approach for context-aware recommendations, in many cases outperforming state-of-the-art CARS algorithms in the Top- N recommendation task.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information filtering

Keywords

Recommendation; Context; Context-aware recommendation; SLIM; Matrix Factorization

1. INTRODUCTION AND RELATED WORK

Context-aware recommender systems (CARS) are an extension of traditional recommender systems (RS) that adapt their recommendations to users' specific situations. The fundamental assumption of CARS is that a rating for an item is a function not just of the user and the item but also of the context in which the item is evaluated or used. A user's preferences on a given item may vary from context to context. For example, *companion* is an influential contextual variable in the movie domain; you may make a different decision in selecting movies if you intend to see the movie on a date, than if you plan on watching a movie with children.

Recent work has demonstrated that matrix factorization (MF) is an effective approach in recommendation. Extensions of standard matrix factorization approach have incorporated contextual

variables, such as time-aware MF. Karatzoglou et al. [3] proposed to use tensor factorization (TF) to integrate context in recommendation. Their approach assumed that contexts are independent of other dimensions. But the computation cost in TF is prohibitive, increasing exponentially as the number of contextual variables increases. Another approach, context-aware matrix factorization (CAMF) [2] was developed to adapt to contextual recommendations by modeling contextual dependencies with the user or item dimensions. CAMF is based on traditional MF [4], where both users and items are represented by factor weightings in a matrix. Those CARS algorithms based on MF have been demonstrated to outperform the ones based on neighborhood collaborative filtering (e.g. differential context modeling [9, 10]).

Most recently, a new matrix factorization for traditional RS, called *sparse linear method (SLIM)* [5] was proposed and shown to be more accurate than the state-of-the-art MF algorithms in the top- N recommendation task. In this paper, we extend this work to create a contextual SLIM (CSLIM) model incorporating contextual conditions. We compare the performance of our proposed approaches to the state-of-the-art context-aware recommendation algorithms through extensive empirical evaluations on multiple data sets.

2. SLIM RECOMMENDER

Assume that there are M users, N items, and let us denote the associated 2-dimensional rating matrix by R . We use u_i to denote user i and t_j to denote the item j . An entry, $R_{i,j}$, in matrix R represents u_i 's rating on t_j . A row vector in R representing u_i 's ratings over all items is denoted by $R_{i,:}$, and $R_{:,j}$ denotes a column vector in R representing ratings on item t_j from all users.

Sparse linear method (SLIM) is an approach designed for top- N recommendations. It improves upon the traditional item-based K -nearest neighbor (ItemKNN) collaborative filtering by learning, directly from the data, a sparse matrix of aggregation coefficients that are analogous to the traditional item-item similarities [5].

In SLIM, the ranking score for user u_i on item t_j is represented by $\hat{S}_{i,j}$, which is calculated by a sparse aggregation of the ratings on the other items that have been rated by u_i . We denote this approach as SLIM-I, since it assigns a N by N non-negative matrix W to represent the aggregation coefficients between each two items. Accordingly, $\hat{S}_{i,j}$ can be estimated by Equation 1.

$$\hat{S}_{i,j} = R_{i,:}W_{:,j} = \sum_{\substack{h=1 \\ h \neq j}}^N R_{i,h}W_{h,j}, \text{ where } \text{diag}(W) = 0 \quad (1)$$

An example of how the SLIM-I approach works is provided in Figure 1. To estimate the ranking score of u_2 on item t_1 , SLIM extracts u_2 's ratings on the other items, and aggregates the ranking score by multiplying of those ratings by the corresponding coeffi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. *RecSys '14*, October 6–10, 2014, Foster City, Silicon Valley, CA, USA. Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...\$15.00. <http://dx.doi.org/10.1145/2645710.2645756>.

coefficients between the item t_1 and t_h (i.e. t_h belongs to the set of other items u_i has rated) in matrix W . The estimated ranking score will be used to rank the items to finally provide top- N recommendations. Similarly, a SLIM-U approach can be easily derived, where $\hat{S}_{i,j}$ is calculated by a sparse aggregation of ratings on this specific item t_j but were given by other users, where W is a M by M non-negative matrix representing the coefficients between every two users. A similar example can be viewed in Figure 2.

	Matrix R	t_1	t_2	t_3	...	t_N		Matrix W	t_1	t_2	...	t_N	
$\hat{S} =$	U_1	0	2	0	...	2	X	t_1	0		
	U_2	?	3	2	...	0		t_2	...	0	
	0
	U_M	0	4	2	...	2		t_N	0	...

Figure 1: Example of SLIM-I

	Matrix R	t_1	t_2	t_3	...	t_N		Matrix W	U_1	U_2	...	U_M
$\hat{S} =$	U_1	0	2	0	...	2	X	U_1	0	
	U_2	?	3	2	...	0		U_2	...	0
	0	...
	U_M	0	4	2	...	2		U_M	0

Figure 2: Example of SLIM-U

Traditional matrix factorization techniques, such as those proposed by Koren et al. [4], usually represent users and items by a set of weights on individual latent factors, and both the number of latent factors and training iterations are required to tune the model and find the optimal solution. In contrast to those approaches, the sizes of matrices R and W are fixed and there is only a single matrix, W , required to be learned in SLIM-I and SLIM-U.

3. CONTEXTUAL SLIM

Analogously to context-aware matrix factorization techniques [2], we propose to incorporate context into the SLIM approach for top- N recommendations. We refer to this as the contextual SLIM (C-SLIM) approach. In the following, we use *contextual factor* to refer to the contextual variables such as "Time", "Location", and "Companion". The term *contextual condition* refers to a specific value in a contextual factor. For example, "weekend" and "weekday" are two contextual conditions for the "Time" contextual factor.

Assume there are F contextual factors and L contextual conditions in total in a context-aware data set, in addition to the M users and N items. The primary task in the CSLIM approach is to estimate the ranking score $\hat{S}_{i,j,c}$ for user u_i on item t_j in context c . We use a binary vector, $c = \langle c_1, c_2, \dots, c_L \rangle$, to denote the contextual situation. For example, assume all contextual conditions with $L=4$ can be represented by {Time=*weekend*, Time=*weekday*, Location=*school*, Location=*home*}. Then, the vector $c = \langle 1, 0, 1, 0 \rangle$ indicates that the current contextual situation is {Time=*weekend*, Location=*school*}.

In CARS, users' preferences may vary from context to context for same item. Therefore, it is necessary to make predictions based on rating profiles in the same context c . Recall that the estimated ranking score matrix \hat{S} in SLIM-I is derived based upon the idea in ItemKNN – aggregation of users' ratings on other items. As a result, it is possible to estimate the ranking score $\hat{S}_{i,j,c}$ by an aggregation of u_i 's ratings on other items in the same contexts c . However, a context-aware data set, with multiple ratings for each item, is usually very sparse – it is not guaranteed that a user has rated other items in the same context c . In this case, we estimate u_i 's rating on an item t_j in context c (i.e. $\hat{R}_{i,j,c}$) based on the user's non-contextual rating on this item (i.e. $R_{i,j}$, ratings without considering contexts) and the aggregated contextual rating deviations (CRD, i.e., the rating deviations in different contextual conditions).

Accordingly, we can build different CSLIM models based on how to estimate the CRD.

3.1 CSLIM-I Models

In this section, we present three CSLIM-I models based on how to estimate the CRD. First, we estimate contextual ratings (i.e. $\hat{R}_{i,j,c}$) based on the non contextual rating $R_{i,j}$ and aggregated CRD. Specifically, we model CRD as the contextual rating deviations on items. In other words, we assume there is a rating deviation for each <item, context condition> pair. Thus, the predicted contextual rating $\hat{R}_{i,j,c}$ can be computed as follows:

$$\hat{R}_{i,j,c} = R_{i,j} + \sum_{l=1}^L D_{j,l}c_l \quad (2)$$

where R is the 2D non-contextual rating matrix. Typically, users may provide non-contextual ratings in addition to the ratings for an item in given contexts. But it is not necessary that both non-contextual ratings and contextual ratings are assigned to a same <user, item> entry. The average rating of each user on each item rated within multiple contexts can be added to the rating matrix R if there is no knowledge about the non-contextual ratings.

We build a $N \times L$ CRD matrix D where each row represents an item, and each column represents an individual contextual condition as shown in Figure 3. Thus, $D_{j,:}$ is a row in D representing CRD for the item t_j in L different contextual conditions.

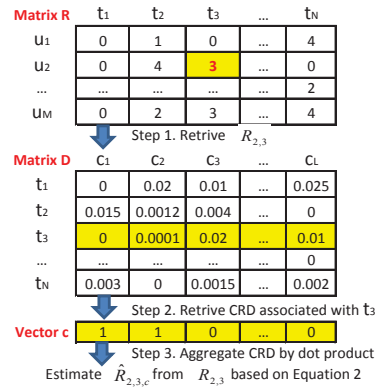


Figure 3: Example: Estimation of Contextual Rating

We can then use the SLIM-I approach to estimate the ranking score $\hat{S}_{i,j,c}$ for user u_i on item t_j in contexts c . This is described in Equation 3, where t_h belongs to the set of items for which user u_i has provided non-contextual ratings (i.e. $R_{i,h} > 0$). The ranking score is estimated by an aggregation of user's ratings on other items in the same context c . We call this approach CSLIM-I-CI because the CRD are assumed for each <item, context condition> pair, and D is built as a CI matrix (i.e. columns and rows represent contexts and items respectively).

$$\hat{S}_{i,j,c} = \sum_{\substack{h=1 \\ h \neq j}}^N \hat{R}_{i,h,c} W_{h,j} = \sum_{\substack{h=1 \\ h \neq j}}^N (R_{i,h} + \sum_{l=1}^L D_{h,l}c_l) W_{h,j} \quad (3)$$

It is important to note that we set $h \neq j$ to avoid the model learning from u_i 's non-contextual rating on t_j . Generally, the CSLIM-I-CI model will learn the parameters in D and W based on each entry of known contextual ratings $R_{i,j,c}$ in the training set, and make predictions only relying on three matrices: non-contextual rating matrix R , CRD matrix D , and the aggregation coefficient matrix W . Using squared error as the optimization criteria, the loss function with regularization terms can be described in Equation 4, which can be solved by stochastic gradient descent (SGD) approach. ∂ and β parameters are the learning rates. Both ℓ_F terms

(e.g. $\|W\|_F^2$) and ℓ_1 terms (e.g. $\|W\|_1$) are included, where the ℓ_1 regularization term is usually applied for sparse models.

$$\underset{D,W}{\text{Minimize}} \frac{1}{2} \left\| R_{i,j,c} - \hat{S}_{i,j,c} \right\|_F^2 + \frac{\beta_1}{2} \|D\|_F^2 + \frac{\beta_2}{2} \|W\|_F^2 + \partial_1 \|D\|_1 + \partial_2 \|W\|_1 \quad (4)$$

In CSLIM-I-CI, D is modeled as a CI matrix, where the deviation is computed for each <item, context condition> pair. The deviations can also be considered for users – we assume there is a rating deviation for each <user, context condition> pair, where D results in a CU matrix (i.e. columns denote contexts and rows represent users). The latter model is called CSLIM-I-CU. In addition to these two approaches, the rating deviations can also simply be viewed as the deviations only associated with each contextual condition rather than as being paired with users or items. In this case, the CRD can be represented by a L -length vector d instead of a matrix D . We call this approach CSLIM-I-C. Note that the only difference among those three CSLIM-I models is the matrix D , while the matrix W is still the same – a $N \times N$ non-negative matrix of item-item coefficients.

3.2 CSLIM-U Models

Contexts can also be incorporated into the SLIM-U models. In this case, W is a $M \times M$ matrix representing user-user coefficients. The matrix D can be built using a similar approach as introduced in the CSLIM-I models. In other words, we can build three CSLIM-U models: CSLIM-U-CI, CSLIM-U-CU, CSLIM-U-C. The CSLIM-U-CU model, for example, utilizes user-based K -nearest-neighbor (UserKNN) collaborative filtering to estimate $\hat{S}_{i,j,c}$ using a sparse aggregation of the ratings on item t_j in contexts c by other users who have rated it. Meanwhile, the CRD matrix D is modeled by a CU matrix which is of size $M \times L$, and W is a $M \times M$ non-negative matrix, with zero values in the diagonal, which is used to estimate coefficients between each pair of users. The loss function is the same as Equation 4 and the estimation for ranking score can be computed by Equation 5, where u_h belongs to the set of users who have rated item t_j except u_i . The parameters in matrices W and D can be updated in a similar way using SGD as the optimizer.

$$\hat{S}_{i,j,c} = \sum_{\substack{h=1 \\ h \neq i}}^M \hat{R}_{h,j,c} W_{i,h} = \sum_{\substack{h=1 \\ h \neq i}}^M (R_{h,j} + \sum_{l=1}^L D_{h,l} c_l) W_{i,h} \quad (5)$$

In a summary, CSLIM-I and CSLIM-U models utilize the intuitions behind ItemKNN and UserKNN, respectively, with the CRD matrix computed in one of three ways: CRD associated to each individual contextual condition, CRD associated with each <item, contextual condition> pair, or CRD associated with each <user, contextual condition> pair, which results in six CSLIM models.

3.3 Pros and Cons

In addition to the CRD matrix D , context-aware matrix factorization (CAMF) [2] has to also learn user and item matrices, as well as other parameters, such as user or item biases. In CSLIM, however, there is only one matrix W must be learned. The complexity of CSLIM models is associated with the number of users or items and the number of contextual conditions. For large-scale recommendations, contexts can be selected in advance based on contextual relevance [6], and feature selection [5] can also be applied to reduce the number of users or items, which was suggested in the original SLIM approach in order to deal with large W matrix.

Moreover, CSLIM helps alleviate the sparsity problem in contextual ratings by making use of the non-contextual ratings. In CSLIM, the contextual ratings are used to train the model, and the ranking score is estimated by the non-contextual ratings and the D

and W matrices. The model can be trained and built incrementally with contextual ratings. It is a good solution for an application that has just started collecting context information.

CSLIM are based on co-ratings among items or users. Thus, this approach may be affected by the cold-start problem. For example, consider a situation where a user has no ratings or an item has not been rated before. This, of course, is a long-standing problem in recommender systems research. One possible solution is to use the item’s or the user’s average rating or even the global average rating in the data to fill the non-contextual rating matrix and then make predictions based on CSLIM models. Once a user has at least one contextual rating, CSLIM is able to learn the corresponding CRD to make recommendations in the future.

4. EXPERIMENTAL EVALUATIONS

4.1 Data Sets

We have evaluated our CSLIM models in five data sets¹. For the reason of limited space, we just present the experimental results for three: *Food* [7], *Restaurant* [8] and *Music* [1] data sets. A summary of those data sets can be viewed in Table 1. For more information, please refer to the papers cited above.

Table 1: Context-aware Data Sets

	Food	Restaurant	Music
Non-contextual Ratings	N/A	50 users, 40 items 896 ratings	42 users, 140 items 930 ratings
Contextual Ratings	212 users, 20 items 6360 ratings	50 users, 40 items 1421 ratings	42 users, 140 items 3010 ratings
# of factors, F	2	2	8
# of conditions, L	7	5	34
Density	$D_{cr}=12.5\%$ $D_{mr}=100\%$	$D_{cr}=11.8\%$ $D_{mr}=39.6\%$	$D_{cr}=5.7e-6$ $D_{mr}=100\%$
Rating Scale	1-5	1-5	1-5

We use two density metrics (see Equation 6) to describe the level of sparsity in the data. Assume there are F contextual factors, D_{cr} measures the density of contextual ratings, where $|\bullet|$ is used to measure the size of the user, item or context dimension. D_{mr} measures the density of multiple ratings in contexts based on unique <user, item> entries. $D_{mr} = 100\%$ indicates each unique <user, item> entry was given multiple ratings (i.e. more than one rating) in different contexts. Thus the Food and Music data are dense in multiple contextual ratings. Food data has no non-contextual ratings, so we use user’s average rating on each item to fill the non-contextual rating matrix R . It is reliable, because $D_{mr} = 100\%$ in this data. Even if D_{mr} is small, we can still use average ratings to fill R . Because the diagonal values in W are 0s, the model will never make a prediction based on user’s existing ratings on the item we’d like to predict.

$$D_{cr} = \frac{\# \text{ of contextual ratings}}{F}, D_{mr} = \frac{\# \text{ of entries was rated multiple times}}{\# \text{ of unique entries in total}} \quad (6)$$

$$\frac{|U||I|}{\prod_{i=1}^F |C_i|}$$

4.2 Evaluation Protocols

We compare the CSLIM algorithms with state-of-the-art context-aware recommendation algorithms, including TF [3], CAMF (i.e. CAMF-CI, CAMF-CU, CAMF-C) [2] and context-aware splitting approaches (CASA) using biased MF as the recommender (i.e. item splitting, user splitting and UI splitting) [11]. To better present the results based on CAMF and CASA, we just present the results by the best performing CASA and CAMF approaches. In addition, we include the original non-contextual SLIM algorithm.

Since SLIM was built for top- N recommendations, we evaluate CSLIM by comparing with baseline algorithms in the top- N

¹Context-aware Data Repository, <http://tiny.cc/contextdata>

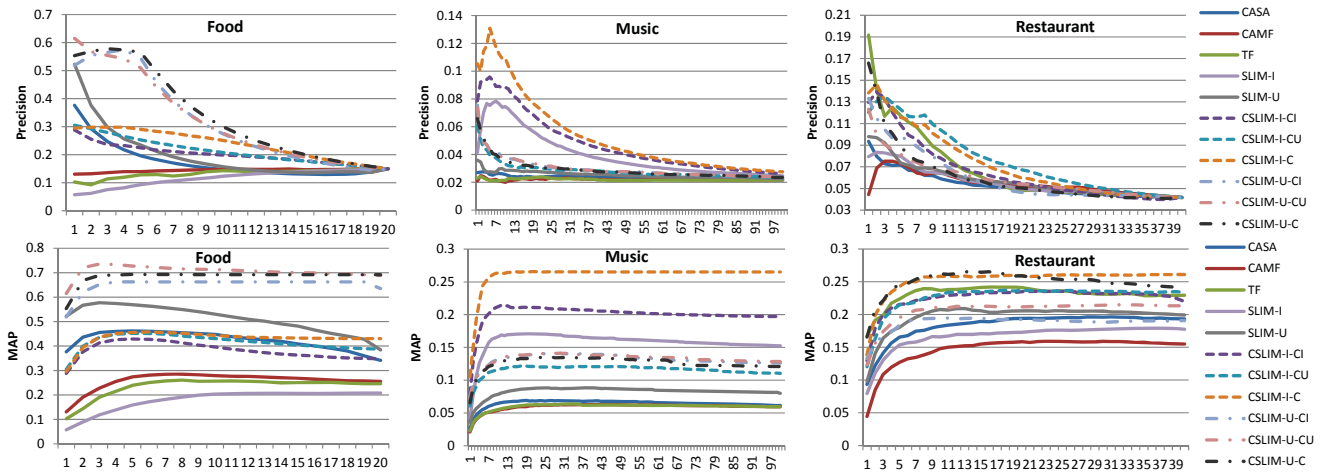


Figure 4: Results in Precision and MAP @ Top- N Recommendations (X-axis represents N ; figures share same legends on the right)

recommendation task. We split the contextual ratings to 5 folds and measure the quality of recommendations using 5-fold cross validation across three metrics: precision, recall and mean average precision (MAP). Those metrics are measured using context as a parameter. Traditionally, precision is measured based on the hit ratio of relevant items in the top- N recommendation list. In CARS, we prefer to add contexts as additional constraints. Specifically, we provide a list of top- N recommended items for each \langle user, contexts \rangle pair, instead of a single user. Then, we measure the hit ratio averaged by each context for each user. This type of evaluation has been used in prior research [11]. Recall and MAP can be measured in a similar manner. The resulting metric values are typically much smaller than the ones in traditional RS, because it is not very common for users to rate multiple items within a same context.

4.3 Experimental Results

The results in precision and MAP are shown in Figure 4. We did not show results for recall, since they are consistent with the ones in precision. As the results indicate, CSLIM approaches are able to significantly outperform the three baseline approaches (i.e. TF, CAMF, CASA), especially in the Food and Music data sets. Overall, the best performing CSLIM provides 102%, 23.7% and 320% improvement on precision@10 for the Food, Restaurant and Music data respectively, compared with the best performing baselines.

The original SLIM algorithms are able to outperform other baseline algorithms (i.e. CASA, CAMF, TF) in some cases, e.g. SLIM-I outperforms the other baseline algorithms in the music data. However, the best performing CSLIM always outperform the original SLIM algorithms in those context-aware data sets.

The ranking of the recommended items is another way of evaluating top- N recommendations, because a high-precision is not necessary to guarantee a good ranking. In terms of MAP, the results show a more significant gap between CSLIM and the baseline approaches. Some baseline approaches are able to outperform some CSLIM approaches in MAP. For example, CASA outperforms the CSLIM-I models in the Food data, and TF works better than some CSLIM-U models in the Restaurant data. But, we observe that the best performing CSLIM models can generally outperform the state-of-the-art context aware recommendation algorithms. Similar observation can be made in the other two data sets we evaluated, however, we did not present the results here due to limited space.

In our experiments, we also find that the performance of CSLIM is correlated with the D_{mr} value. A large D_{mr} value indicates users have multiple ratings on items in different contexts, where

the parameters in both the CRD matrix D and coefficient matrix W will be learned in multiple times based on those dense entries and it is able to finally result in a more accurate ranking model for top- N context-aware recommendations.

5. CONCLUSION AND FUTURE WORK

In this paper, we successfully incorporate contexts into SLIM and develop several CSLIM algorithms, where they estimate the ranking score by the intuition behind ItemKNN or UserKNN via the aggregation of users' estimated contextual ratings on items. CSLIM models are demonstrated to outperform the state-of-the-art CARS algorithms for top- N recommendations in our experiments. We also find that the performance of CSLIM is correlated with the density of multiple ratings in contexts. In our future work, we plan to incorporate the contexts into the coefficient matrix W , which helps estimate the coefficients between each contextual conditions and discover more insights about the contextual effects.

6. REFERENCES

- [1] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lükke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies*, pages 89–100. Springer, 2011.
- [2] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *ACM RecSys*, pages 301–304, 2011.
- [3] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *ACM RecSys*, pages 79–86, 2010.
- [4] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [5] X. Ning and G. Karypis. Slim: Sparse linear methods for top-n recommender systems. In *IEEE ICDM*, pages 497–506. IEEE, 2011.
- [6] A. Odic, M. Tkalcic, J. F. Tasic, and A. Košir. Relevant context in a movie recommender system: Users' opinion vs. statistical detection. *RecSys Workshop on Context-Aware Recommender Systems*, 2012.
- [7] C. Ono, Y. Takishima, Y. Motomura, and H. Asoh. Context-aware preference model based on a study of difference between real and supposed situation data. In *User Modeling, Adaptation, and Personalization*, pages 102–113, 2009.
- [8] X. Ramirez-Garcia and M. Garca-Valdez. Post-filtering for a restaurant context-aware recommender system. In *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, volume 547, pages 695–707. Springer, 2014.
- [9] Y. Zheng, R. Burke, and B. Mobasher. Differential context modeling in collaborative filtering. In *Proceedings of School of Computing Research Symposium*. DePaul University, Chicago IL, USA, 2013.
- [10] Y. Zheng, R. Burke, and B. Mobasher. Recommendation with differential context weighting. In *The 21st Conference on User Modeling, Adaptation and Personalization*, pages 152–164, 2013.
- [11] Y. Zheng, R. Burke, and B. Mobasher. Splitting approaches for context-aware recommendation: An empirical study. In *ACM SAC*, pages 274–279, 2014.