Context-Aware Recommender Systems: A Comparison Of Three Approaches

Umberto Panniello and Michele Gorgoglione

Politecnico di Bari (Italy), Viale Japigia 182

Tel. +39 080 5962765

Fax: +39 080 5962766

u.panniello@poliba.it, m.gorgoglione@poliba.it

Abstract. Methods for generating context-aware recommendations were classified into the pre-filtering, post-filtering and contextual modeling approaches. This paper proposes a novel type of contextual modeling, that is called *contextual neighbors*, based on the idea of using context to compute the neighborhood in a collaborative filtering approach, and introduces four variants of this method. In addition, the paper presents the results of the comparison among these four approaches and among the *contextual neighbors* approach to the other contextual approaches and to the un-contextual one. While some of these methods have been studied independently, few prior research has compared their performance to determine which of them is better.

1 Introduction

Most traditional Recommender Systems (RSs) provide recommendations of items to users and vice versa and do not take into consideration the circumstances and other contextual information when recommendations take place. For example, when an online travel agency recommends a vacation package, it is important to know when the person plans to go on vacation. Recently, some companies started taking into account the contextual information (i.e., Sourcetone). In academia, several studies, such as [1], demonstrated that context induces important changes in a customer purchasing behavior. Experimental research on customer modeling suggests that including context in a customer behavior model improves the ability to predict her behavior in some cases because it allows the identification of more homogeneous patterns in the data describing the purchasing history of a customer [2]. Therefore, the accuracy of predicting consumer preferences should depend on the degree to which we have incorporated the relevant contextual in-formation. The usage of contextual information in context-aware recommender systems (CARS) can be broadly classified into two groups: (1) recommendation via context-driven querying and search, and (2) recommendation via contextual preference elicitation and estimation. The approach based on the context-driven querying and search has been used by several mobile and tourist recommender systems, e.g., [3], [4], that typically use contextual information to query or search a certain repository of resources (e.g., restaurants) and present the best matching resources (e.g., nearby restaurants that are currently open) to the user.

The other approach to CARS, that we follow in this paper, is based on contextual preference elicitation and estimation, e.g., [5], [6], [7,8], including elicitation and estimation of ratings of various items provided by different users. This approach can be traced back to the work in [6] and [9]. [9] hypothesized that the inclusion of knowledge about the user's task into the recommendation algorithm in certain applications can lead to better recommendations. [6] described a way to incorporate the contextual information into recommender systems by using a multidimensional approach in which the traditional 2-dimensional (2D) User/Item paradigm was extended to support additional contextual dimensions, such as Time, Location and Company. Since then, several contextual preference elicitation and estimation approaches to CARS have been proposed, all of them emphasizing the need to model and learn user's context-sensitive preferences. Many of these methods are reviewed in [10] and [7]. Once the context-sensitive preferences of users are learned, recommendations are generated by either adapting the existing collaborative filtering, content-based, or hybrid recommendation methods to context-aware recommendation settings or by developing novel intelligent data analysis techniques from data mining and machine learning. Furthermore, several scholars [5], [7,8] have shown that adding contextual information helps to improve estimations of unknown ratings in this approach. For example, [5] and [6] described a way to include the contextual information by using a multidimensional approach, as it was mentioned above. Also, it was shown in [5] that the multidimensional contextual information does matter and can lead to better recommendations in comparison to the traditional un-contextual 2D recommender systems.

2 Pre-filtering, post-filtering and contextual modeling

In this paper, we use implicit ratings following the approach outlined by [11], [12], [13]. We measure the utility of product j for user i with the purchasing frequency x_{ij} specifying how often user i purchased product j. Unlike the traditional two-dimensional (2D) recommender systems that try to estimate unknown ratings in the $Users \times Items$ matrix, where *Users* and *Items* are the sets of users and items respectively, context-aware recommender systems (CARS) also take into account contextual information in a $Users \times Items \times Context$ matrix. *Context* is a set of contextual attributes C each having a hierarchical structure defined by a set of q atomic attributes, i.e., $C = (C_1,\ldots, C_q)$ [10]. Further, the values taken by attribute C_q define finer (more granular) levels, while C_1 coarser (less granular) levels of

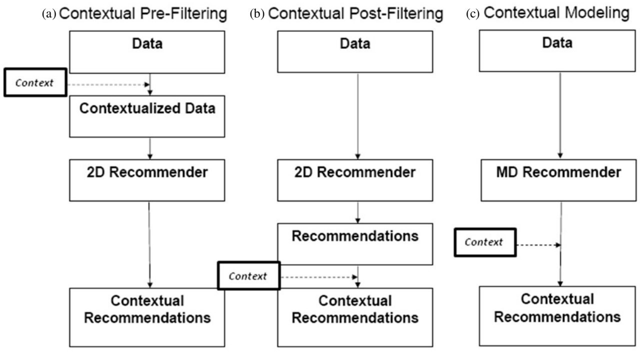

Fig. 1 How to use context in the recommendation process

contextual knowledge [14]. For example, Fig. 2 presents the hierarchy for the contextual attributes used in our analysis (see Section 3).

In this paper we compare the three contextual approaches proposed in [10] (also shown in **Errore. L'origine riferimento non è stata trovata.**), that start with "Data" on users, items, ratings and contextual information ("Context") and results in generating context-specific recommendations, are presented. In the contextual pre-filtering (PreF) the contextual information is used to filter out irrelevant ratings before they are used for computing recommendations with standard (2D) methods. In the contextual post-filtering (PoF) context is used after the classical (2D) recommendation methods are applied to the standard recommendation data. In the contextual modeling (CM) context is used inside the recommendation-generating algorithms. The work presented in [10] helped researchers to understand different aspects of using the contextual information in the recommendation process. However, [10] did not examine which of these methods are more effective for providing contextual recommendations, and therefore, left this important topic unaddressed and without any prescriptive recommendations for which method to use. In this paper we empirically compare the three contextual approaches, i.e., pre-filtering, post-filtering and contextual modeling among themselves to determine which one is better, and also with the un-contextual approach when contextual information is ignored and not used in the recommendation methods. We also present several types of contextual post-filtering and contextual modeling methods and evaluate their performance in order to identify the best-performing methods in these two categories. We show that one particular post-filtering method dominates other approaches. However, this dominance comes with a certain complexity "cost" that prevents that method to be a clear winner in all the practical settings.

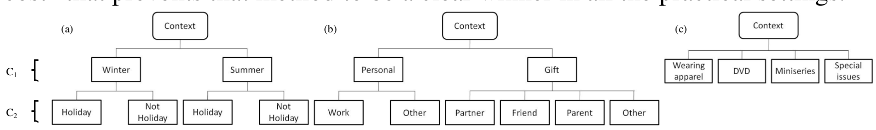

Fig. 2 Hierarchical structure of the contextual attributes (a) Time Of The Year, (b) Intent Of Purchase and (c) Store

2.1 The pre- and the post-filtering approaches

According to the pre-filtering approach (**Errore. L'origine riferimento non è stata trovata.**(a) and [10]), the contextual information is used as a label for filtering out those ratings that do not correspond to the specified contextual information. This filtering is done *before* the main recommendation method is launched on the remaining data that passed the filter. In other words, if a particular context of interest is *C*, then the pre-filtering method selects from the initial set of all the ratings only those corresponding to the specified context *C*. As a result, it generates the *Users* $\times$ *Items* matrix containing only the data pertaining to context *C*. Then the 2D recommendation method (e.g., collaborative filtering) is launched on this remaining dataset that passed the filter to generate recommendations for context *C*. We call this approach Exact Pre-Filtering (*EPF*). According to the post-filtering (PoF) approach (see **Errore. L'origine riferimento non è stata trovata.**(b) and [10]), we first ignore all the contextual information in the data and apply a traditional 2D recommendation method, such as collaborative filtering, on the *whole un-contextual* data set, where the contextual information is dropped. Once the unknown ratings are estimated using the 2D method on this data and the un-contextual recommendations are produced, we "contextualize" these recommendations as follows. Although there exist various methods for contextualizing the 2D recommendations [10], in this paper we consider the following two approaches, called "*Weight*" and "*Filter*". Both approaches analyze data for a given user in a given context to calculate the probability with which the user chooses a certain item in the given context. After that, the recommendations obtained using this 2D method are "contextualized" by using the contextual probabilities. For detailed information on the methods, see [15].

2.2 The contextual modeling approach

According to the contextual modeling approach, we present a new method called *contextual-neighbors* CM. This approach is based on user-based collaborative filtering (CF) and works as follows. First, for each user *i* and context *c*, we define the user profile in context *c*, i.e. the *contextual profile Prof(i, c)*. For example, if the contextual variable *c* has two values (e.g., Winter and Summer), then we have two contextual profiles for each user, one for the Winter and the other for the Summer. Note that these contextual profiles can be defined in many different ways, some of which are presented in [2], and our approach does not depend on any particular choice of a profiling method. However, in the experimental study described in Section 4 we use the following specific contextual profiling technique. As explained in Section 2, we follow the transaction-based approach to RSs and measure the utility r_{ijc} of product *j* for user *i* in context *c* with the purchasing

frequency x_{ijc} specifying how often user i purchased product j in context c. Then we use this measure to define contextual profile as $Prof(i, c) = (r_{i1c}, \ldots, r_{ikc})$.
We use these profiles to define similarity among users and also to define and find N nearest "neighbors" of user i in context c, where "neighbors" are determined using contextual profiles $Prof(i', c')$ and similarity measures between the profiles. Although this similarity can be defined in various ways, we use a popular CF approach and define the similarity measure d using the cosine measure as:

$$d(Prof(i',c'), Prof(i,c)) = \frac{Prof(i,c) \bullet Prof(i',c')}{\|Prof(i,c)\|^2 \times \|Prof(i',c')\|^2} = \frac{\sum_{s \in S_{ii'c}} r_{isc} r_{i'sc'}}{\sqrt{\sum_{s \in S_{ii'c}} r_{isc}^2} \sqrt{\sum_{s \in S_{ii'c}} r_{i'sc'}^2}} \quad (1)$$

where $r_{isc'}$ and $r_{i'sc'}$ are the ratings of item s assigned by user i and user i' respectively in context c and c'. $S_{ii'c}=\{s \in Items \mid r_{isc} \neq \varnothing \wedge r_{i'sc'} \neq \varnothing\}$ is the set of all items co-rated by both user i and user i' in context c.
Then we find N nearest neighbors for the (i, c) pair by identifying pairs (i', c') such that $d(Prof(i', c'), Prof(i, c))$ is the largest among all the candidate pairs (i', c') subject to the following constraints:

- Mdl_1: there are no constraints on the set of (i', c') pairs, and we select N pairs that are the most similar to (i, c).
- Mdl_2: we select an equal proportion of pairs (i', c') corresponding to each context c (e.g., if the contextual variable has only two values, Winter and Summer respectively, and the neighborhood size is 80, we select 40 neighbors from Winter and 40 from Summer).
- Mdl_3: we select N pairs (i', c') that are the most similar to (i, c) corresponding to each context c at the same level of the context of interest (e.g., if the context of interest is "Winter Holiday" in Fig. 2(a), we select the neighborhood by using only profiles referred to level C_2 of that contextual variable).
- Mdl_4: we select an equal proportion of pairs (i', c') corresponding to each context c at the same level of the context of interest (e.g., if the context of interest is "Winter Holiday" in Fig. 2(a) and the neighborhood size is 80, we define the neighborhood by using 20 users from the context "Winter Holiday", 20 users from the context "Winter Not Holiday", 20 users from the context "Summer Holiday" and 20 users from the context "Summer Not Holiday").

3 Experimental Setup

We compared the pre-filtering, post-filtering, contextual modeling and un-contextual recommendations across a wide range of experimental settings. First, we selected three different data sets. The first dataset (DB1) comes from an e-commerce website commercially operating in an European country which sells electronic products to approximately 120,000 users and contains about 220,000 purchasing transactions during an observation period of three years. For this dataset, we selected the time of the year as a contextual variable. Its hierarchical structure is presented in Fig. 2(a). The classification into Summer or Winter and Holiday or Not Holiday is based on the experiences of the CEO of the e-commerce website that we used in our study. The data was pre-processed by excluding customers who made only one single transaction or had abnormal behavior or had transactions either only in the first two years or only in the third year. The resulting dataset contained about 1,500 users and about 10,000 transactions thus having about 7 transactions per user.

The second dataset (DB2) is taken from the study described in [2]. First, a special purpose browser was developed to help users navigate Amazon.com website and purchase products on its site. This browser was made available to a group of students who were asked to navigate and simulate purchases on Amazon.com during a period of four months. Once a product was selected by a student to be purchased, the browser recorded the selected item, the purchasing price and other useful characteristics of the transaction and this information was stored in the database. In addition, the student was asked at the beginning of each browsing session to specify its context, which was the intent of a purchase in our case. The structure of this contextual variable IntentOfPurchase is presented in Fig. 2(b). Further, the data was pre-processed by excluding the students who made less than 40 transactions and eliminating the students who had any kind of misleading or abnormal behavior. The resulting number of students was 556, and the total number of purchasing transactions for the students was 31,925 thus having about 57 transactions per user.

The third dataset (DB3) comes from an e-commerce website which sells comics and comics-related products, such as T-shirts, DVDs and gadgets. It contains about 50,000 transactions and 5,000 users thus having about 10 transactions per user. We used the store (i.e., the section in the Web site where products are bought), as a contextual variable (see Fig. 2(c)). The importance of this contextual variable comes from the expectation that customers' behavior changes when navigating and buying products in different sections of the Web site.

In our study, we recommend product categories instead of individual items because the e-commerce applications that we consider have very large numbers of items (hundreds of thousands or even millions). We tried different item aggregation strategies and found that the best results are for 14 categories for DB1, 24 categories for DB2 and 136 for DB3. We aggregated items into categories according to the classification provided by the Web sites product catalogue. Estimations of unknown utilities were done by using a standard user-based collaborative filtering (CF) method [16]. According to the CF approach, the neighborhood was formed using the cosine similarity [17]. The neighborhood size N was set to $N = 80$ users, which proved to be the optimal size in our experiments.

The experiments were performed for datasets DB1, DB2 and DB3 for all the levels of contextual knowledge (un-contextual, C_1 and C_2), as presented in Fig. 2. We have performed t-tests in order to determine if the chosen contextual variables matter. The results of these tests were statistically significant (at 95% level) and demonstrated that the contextual variables TimeOfTheYear, IntentOfPurchase and Store matter. We used Precision, Recall, F-Measure, Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*) [18] as the performance measures in our experiments, as done in [5]. We computed MAE and RSME in a standard way by taking absolute and squared differences between the estimated and actual values of estimated utilities since the utilities are measured using discrete variables [19], [20], [21]. Finally, we divided each dataset into the training and the validation sets, the training set containing 2/3 and the validation set 1/3 of the whole dataset. For the DB1 dataset, the first two years were the training set and the third year was the validation set. For the DB2 dataset, we randomly split it in 2/3 for the training set and the remaining 1/3 for the validation set (in this case, it was impossible to make a good temporal split because all the transactions were made within a couple of months). For the DB3 dataset, the first nine months were the training set and the last three months were the validation set.

4 Results

Firstly, we compare the traditional un-contextual 2D collaborative filtering and the exact pre-filtering (*EPF*) methods. This type of comparison has been studied in [5] before, where it was shown that in certain cases the un-contextual approach dominates, while in other cases the contextual one dominates the un-contextual method. However, the work reported in [5] was done in the context of multi-dimensional recommendations and only for one small dataset. In this study, we wanted to provide a more extensive comparison of the un-contextual and the pre-filtering methods across several and bigger datasets and across numerous other experimental settings. Furthermore, we also needed to conduct this comparison for the uniformity reasons because we also do the comparison of the post-filtering and the contextual modeling approaches to the un-contextual approach in this paper.

We compared the un-contextual and the *EPF* contextual methods for the case of user-based collaborative filtering, across the datasets DB1, DB2 and DB3, across multiple levels of classification hierarchy (C_1 and C_2) and different performance measures. We split the data into the training and testing sets as described in Section 4. For the sake of brevity we only present a summary of these results in Table 1 where only the F-measure, averaged across the experimental setting, is reported and Table 2 where only the MAE, averaged across the experiment settings, is reported. Table 1 and Table 2 report all the accuracy gains (in terms of F-measure and MAE, namely) across each recommender systems for DB1, DB2 and DB3 (negative values mean performance reduction). For example, the first row of Table 1 shows the performance gains (reductions), in terms of F-measure, for the un-contextual RS vis-à-vis the *EPF*, *Filter PoF*, *Weight PoF* and *Mdl1* methods. The matrixes in Table 1 and Table 2 are anti-symmetric, as should be the case when two methods are compared in terms of their relative performance. However, the comparison among the other performance measures used in the experiments is discussed throughout this section.

The *EPF* dominates the un-contextual case in some cases in terms of the F-measure and is dominated in other cases, consistently with [5]. The intuitive explanation is that when the contextual information is very granular, this results in a more homogeneous rating prediction model but also leads to data sparsity issues (only few context-specific ratings are used to build the model). In contrast, when the contextual information is too coarse, we can have enough data to build the model, but it is more heterogeneous. This conflict between homogeneity of the model and rating sparsity produces mixed results when comparing *EPF* and the un-contextual model. Furthermore, *EPF* outperforms the un-contextual model in terms of *MAE* and *RMSE* (see Table 2).

Secondly, we compare the un-contextual and the post-filtering methods. Unlike exact pre-filtering, there exist many post-filtering methods. The comparison can depend very significantly on the choice. We compared the *Weight PoF* and *Filter PoF* and the un-contextual method across various experimental conditions described in Section 3. The *Filter PoF* dominates the un-contextual case across all the levels of context for the F-measure: for DB1, the difference between contextual and un-contextual models is 20% on average, for DB2 it is 27% on average and for the DB3 it is 90%. On the contrary, *Weight PoF* is always dominated by the un-contextual model. All this makes us to conclude that the comparison of the post-filtering and un-contextual methods depends very significantly on the type of the post-filtering method being used. This observation is not surprising because the post-filtering method takes the results of the same 2D un-contextual method and reorders them based on the context-based post-filtering heuristic. If the heuristic is “good” (such as *Filter*), then this re-ordering improves recommendation quality; otherwise, it can make the results even worse (as for heuristic *Weight*).

Thirdly, we compare the un-contextual and the contextual modeling methods. We consider four methods Mdl_1, Mdl_2, Mdl_3, Mdl_4 in our study and compare them to the un-contextual method across the experimental conditions. The experiments proved that the performances of the four CM approaches are very similar and the differences are not statistically significant. For this reason we only present the performance of Mdl_1 because it slightly dominates other CM methods in some cases. This makes sense because the *N* neighbors are selected for Mdl_1 in an unconstrained manner, whereas they are selected based on various constraints for the other three approaches. As Table 1 and Table 2 show, Mdl_1 always outperforms the un-contextual model.

Finally, we compare the performance of the pre-filtering, post-filtering and contextual modeling methods. Table 1 shows that *Weight PoF* is the worst recommender system (it never outperforms any other RS) followed by the un-contextual one (it outperforms only the *Weight PoF* while it is dominated by all the other context-aware RSs). On the other side, the *Filter PoF* is the best recommender system (it outperforms all the other systems and it is dominated by the Mdl_1 only for the DB1) followed by the Mdl_1 (it generally outperforms all the other RSs with the exception of *Filter PoF*). The *EPF* always outperforms the un-contextual recommender system, while in some cases dominates the other context-aware RSs and in other cases is dominated by them. Table 2 shows similar results. In summary, the answer to the question of which contextual approach provides the best performance depends on the type of the post-filtering method used. In fact, although the *Filter PoF* does not outperform all the other approaches in every setting, it does in most settings, while the *Weight PoF* is often outperformed by all the other context-based approaches. Overall, the *Filter PoF* is the best approach in 78% of comparisons in terms of F-measure (i.e., seven times out of 9), 67% in terms of Precision, 100% in terms of *MAE* and *RMSE* (i.e., three times out of three). In terms of F-measure, the CM approach (Mdl_1) is the best in 44% of comparisons (percentage values do not sum to 100% because in some cases approaches are equivalent), while *EPF* only 11%. The alternative post-filtering method, *Weight PoF*, has never the highest performance in terms of F-measure. In contrast, it has the lowest F-measure and Precision in 78% of cases, the lowest Recall in 33% of comparisons, and it is always the worst in terms of *MAE* and Recall.

Although we demonstrated that the post-filtering method *Filter PoF* dominates alternative contextual approaches considered in the paper in terms of better recommendation performance, it does not mean that it should always be used in practice for the following reasons. First, using *Filter PoF* entails finding the right parameters for the method, such as the size of the neighborhood *N* and the threshold value. Determining good values of these parameters in a specific application can be time consuming and expensive. Also, different post-filtering methods, such as *Weight* vs. *Filter*, have different types of parameters, and this complicates the selection of the best post-filtering method even further.

Table 1 F-measure gains (reductions) across recommender systems.

		Un-contextual	EPF	Filter PoF	Weight PoF	Mdl_1
DB1	Un-contextual	0%	-2%	-20%	27%	-22%
	EPF	2%	0%	-19%	29%	-21%
	Filter PoF	20%	19%	0%	59%	-3%
	Weight PoF	-27%	-29%	-59%	0%	-39%
	Mdl_1	22%	21%	3%	39%	0%
DB2	Un-contextual	0%	-2%	-27%	14%	-7%
	EPF	2%	0%	-26%	16%	-5%
	Filter PoF	27%	26%	0%	57%	28%
	Weight PoF	-14%	-16%	-57%	0%	-18%
	Mdl_1	7%	5%	-28%	18%	0%
DB3	Un-contextual	0%	-88%	-90%	8%	-88%
	EPF	88%	0%	-12%	819%	5%
	Filter PoF	90%	12%	0%	942%	19%
	Weight PoF	-8%	-819%	-942%	0%	-89%
	Mdl_1	88%	-5%	-19%	89%	0%

Table 2 MAE gains (reductions) across recommender systems.

		Un-contextual	EPF	Filter PoF	Weight PoF	Mdl_1
DB1	Un-contextual	0%	-57%	-257%	-9%	-88%
	EPF	57%	0%	-127%	31%	-20%
	Filter PoF	257%	127%	0%	70%	47%
	Weight PoF	9%	-31%	-70%	0%	-73%
	Mdl_1	88%	20%	-47%	73%	0%
DB2	Un-contextual	0%	-66%	-214%	-14%	-76%
	EPF	66%	0%	-89%	31%	-6%
	Filter PoF	214%	89%	0%	64%	44%
	Weight PoF	14%	-31%	-64%	0%	-55%
	Mdl_1	76%	6%	-44%	55%	0%
DB3	Un-contextual	0%	-205%	-250%	0%	-233%
	EPF	205%	0%	-15%	67%	-9%
	Filter PoF	250%	15%	0%	71%	5%
	Weight PoF	0%	-67%	-71%	0%	-233%
	Mdl_1	233%	9%	-5%	233%	0%

Finally, post-filtering methods are computationally more expensive than the pre-filtering method *EPF* because only a relatively small subset of data is used in estimations of unknown ratings for *EPF*. Therefore, the selection of the best contextual modeling method in a given application is more complicated in practice than a simple rule “always use *Filter PoF* in all the settings.” The CM approach is a good alternative in practice, because it proved to be the second-best and the performance is stable across various possible CM methods unlike the post-filtering approach.

5 Conclusions

In this paper, we compared the un-contextual and contextual recommender systems, for which we considered the pre-filtering, the post-filtering and the contextual modeling methods of generating contextual recommendations. In particular, we used the exact pre-filtering (*EPF*) and the *Weight* and the *Filter* post-filtering methods for the first two approaches. Moreover, we proposed a new type of contextual modeling, that we called *contextual neighbors* CM, and four specific types of contextual neighbors methods, called Mdl_1, Mdl_2, Mdl_3 and Mdl_4, each of them selecting contextual neighborhoods in a somewhat different way.

We compared the un-contextual with the contextual methods across various experimental settings, including three datasets, different level of item aggregation, different neighborhood sizes, seven recommendation engines (un-contextual, *EPF*, *Filter PoF*, *Weight PoF*, Mdl_1, Mdl_2, Mdl_3 and Mdl_4), different contextual levels

(C_1 and C_2) and several performance measures (Precision, Recall, F-Measure, *MAE* and *RMSE*). We showed that the contextual *Filter* method dominates the un-contextual one and that the un-contextual method dominates the *Weight* method. We also showed that *EPF* dominates the un-contextual method in some cases and is inferior in other cases on the datasets (and the corresponding applications) used in our study. We also compared the contextual neighbors methods to identify the best performing one. Although Mdl_1 slightly outperforms the others, there are no significant performance differences among them. This result is not surprising because different ways of selecting contextual neighborhood do not fundamentally change recommendation results. We have then selected Mdl_1 as the best-of-breed contextual modeling method and compared it with the pre-, post- filtering and un-contextual methods. We showed that Mdl_1 dominates the traditional un-contextual approach and is comparable to the pre-filtering method (*EPF*). We have also shown that Mdl_1 dominates some of the less advanced post-filtering methods (such as *Weight PoF*) but is inferior to the best post-filtering methods (such as *Filter PoF*). This implies, among other things, that, in order to decide which approach should be used in a particular recommendation application, various post-filtering methods should be carefully compared which is a laborious and a time-consuming strategy. However, the CM approach is a good alternative in practice, because it proved to be the second-best and the performance is stable across various possible CM methods unlike the post-filtering approach.
As future work we intend to analyze deeper different aspects of the comparisons between pre-filtering, post-filtering and contextual modeling approaches. In particular, we plan to analyze the impact of different approaches in terms of generated profit and sales, diversity of the recommended items and trust of the users on the different recommender systems. Moreover, we will compare the *contextual neighbor* approach to other contextual modeling approaches.

References

1. Bettman, J., Luce, M., & Payne, J. (1998). Constructive Consumer Choice Processes. *Journal of Consumer Research, 25*(3), 187-217.
2. Palmisano, C., Tuzhilin, A., & Gorgoglione, M. (2008). Using Context to Improve Predictive Modeling of Customers in Personalization Applications. *IEEE Transactions on Knowledge and Data Engineering, 20*(11), 1535-1549.
3. Cena, F., Console, L., Gena, C., Goy, A., Levi, G., Modeo, S., et al. (2006). Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Commun., 19*(4), 369-384.
4. van Setten, M., Pokraev, S., & Koolwaaij, J. (2004). Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In *Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 235-244).

5. Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems, 23*, 103-145.
6. Adomavicius, G., & Tuzhilin, A. (2001). Multidimensional Recommender Systems: A Data Warehousing Approach. In *Electronic Commerce* (pp. 180-192).
7. Anand, S., Mobasher, B., Berendt, B., Hotho, A., Mladenic, D., & Semeraro, G. (2007). Contextual Recommendation. In *From Web to Social Web: Discovering and Deploying User and Content Profiles* (Vol. 4737, pp. 142-160): Springer Berlin / Heidelberg.
8. Yu, Z., Zhou, X., Zhang, D., Chin, C.-Y., Wang, X., & Men, J. (2006). Supporting Context-Aware Media Recommendations for Smart Phones. *Pervasive Computing, IEEE, 5*(3), 68-75.
9. Herlocker, J., & Konstan, J. (2001). Content-Independent Task-Focused Recommendation. *IEEE Internet Computing, 5*(6), 40-47.
10. Adomavicius, G., & Tuzhilin, A. (2010). Context-Aware Recommender Systems. In Springer (Ed.), *Handbook on Recommender Systems*.
11. Chen, L. S., Hsu, F. H., Chen, M. C., & Hsu, Y. C. (2008). Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences, 178*(4), 1032-1048.
12. Huang, Z., Chung, W., & Chen, H. A graph model for e-commerce recommender systems. In *Journal of the American Society for Information Science and Technology, 2004* (Vol. 55, pp. 259-274, Vol. 3)
13. Huang, Z., Li, X., & Chen, H. Link prediction approach to collaborative filtering. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, 2005* (pp. 141-142): ACM Press
14. Kwon, O., & Kim, J. (2009). Concept lattices for visualizing and generating user profiles for context-aware service recommendations. *Expert Systems with Applications, 36*(2), 1893-1902.
15. Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., & Pedone, A. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems, New York, New York, USA, 2009* (pp. 265-268): ACM
16. Resnick, P., & Varian, H. R. (1997). Recommender systems. *Commun. ACM, 40*(3), 56-58, doi:10.1145/245108.245121.
17. Breese, J., Heckerman, D., & Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence 1998* (Vol. 461, pp. 43-52)
18. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5-53, doi:10.1145/963770.963772.
19. Nichols, D. M. Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, 1998* (pp. 31-36)
20. Oard, D., & Kim, J. Implicit Feedback for Recommender Systems In *Proceedings of the AAAI Workshop on Recommender Systems, 1998* (pp. 81-83)
21. Adomavicius, G., Huang, Z., & Tuzhilin, A. (2008). Personalization and Recommender Systems. In Z.-L. C. a. S. Raghavan (Ed.), *State-of-the-Art Decision Making Tools in the Information-Intensive Age* (pp. 55-100): Tutorials in Operations Research.