Context-aware item-to-item recommendation within the factorization framework

Balázs Hidasi[†][*]
balazs.hidasi@gravityrd.com

Domonkos Tikk[†]
domonkos.tikk@gravityrd.com

[†]Gravity R&D
Export tér 5–7.
1101 Budapest
Hungary

[*]Budapest University of Technology and Economics
Magyar Tudósok krt. 2
1117 Budapest
Hungary

ABSTRACT

Item-to-item recommendation – when the most similar items sought to the actual item – is an important recommendation scenario in practical recommender systems. One way to solve this task is to use the similarity between item feature vectors of factorization models. By doing so, one may transfer the well-known accuracy of factorization models observed at the personalized recommendations to the item-to-item case. This paper introduces context-awareness to item similarities in the factorization framework. Two levels of context-aware similarities are defined and applied to two context-aware implicit feedback based factorization methods (iTALS and iTALSx). We investigate the advantages and drawbacks of the approaches on four real life implicit feedback data sets and we characterize the conditions for their application. The results suggest that it is worth using contextual information for item-to-item recommendations in the factorization framework, however, one should carefully select the appropriate method to achieve similar accuracy gain than in the case of the more general item-to-user recommendation scenario.

Categories and Subject Descriptors

I.2.6 [[**Artificial Intelligence**]]: Learning - Parameter Learning

General Terms

Algorithms, Experimentation

Keywords

recommender systems, item-to-item recommendation, implicit feedback, factorization, context-awareness

1. INTRODUCTION

Item-to-item recommendation is a typical recommendation scenario in real-world recommender systems. One can, for instance, partially overcome the user cold-start problem by providing non-personalized item-to-item recommendations relevant to a selected/viewed item for a new user. Items that are most similar to the given item are then displayed. The similarity can be here defined in different ways, using for example metadata ("similar items") or transactional data ("users viewed this item also purchased the followings"). Since the concept of item similarity is an intuitive notion, typically not explicitly defined for the recommender, the quality of the similar items greatly depends on the evaluation (that is adjusted to business requirements).

Item-based collaborative filtering approaches are occasionally also referred to as item-to-item methods. This is because similarity based approaches can be easily transformed into personalized ones (item-to-user) by recommending items that are similar to the ones in the user's history. To avoid confusion, in this paper we use the term *item-to-item recommendation* for the concept of recommending similar items to a given item. Therefore, in our case, the recommendation list does not depend on the visiting user, but only on the visited item.

Collaborative filtering (CF) recommender algorithms use only transactional data, yet considered as the state-of-the-art methodology for personalized recommendation. Factorization based methods compute a low rank vector (feature vector) for each user and each item, and preferences of a user on an item is approximated by the scalar product of the appropriate vectors. In the general item-to-user scenario – where items that satisfy the needs of the user the most are recommended – factorization based models proved to be a scalable and accurate way to tackle the recommendation problem. Therefore, we investigate in this paper how to apply the same framework for item-to-item recommendations using similarity between the *item* feature vectors.

CF methods work on transactional data. Transactional data is often classified into explicit or implicit feedback types. The main differences are that explicit feedbacks directly describe the preferences of the user on items (e.g. in the form of ratings), and both positive and negative preference data are provided by users. On the other hand, implicit feedback data are indirect clues for user preferences; that can only be inferred from user's navigational and purchase history (its elements termed as *events*). For implicit feedback, web shop usage is a typical example. The presence of an event does not always mean positive feedback (e.g. whether the product is

CaRR'13, February 5, 2013, Rome, Italy.

satisfactory only turns out after the purchase) therefore the positive feedback is noisy. But more importantly, the absence of an event hardly means negative preference, because the user is usually not aware of the majority of the available products/content. Thus, with implicit feedback data, information on negative preferences is not available. This property makes preference reasoning more difficult both in term of prediction accuracy and scalability. Nevertheless, the practical usefulness of the implicit feedback based recommenders justifies their necessity.

Context-aware recommender systems integrate contextual information into recommendation approach. In this paper context is used as "event-context", that is, context is considered as a property of a user–item interaction (e.g. time of the event, actual mood of the user when recommendation is requested, etc.) in contrast to static properties specific to users (user metadata) or items (item metadata). Context-aware approaches have substantial advantages over other collaborative filtering methods as they tend to be more accurate and able to adapt to temporal circumstances (like the mood of the user).

This paper deals with the intersection of the above four areas: item-to-item recommendations, context-awareness, implicit feedback data and factorization models. We investigate our recent context-aware tensor factorization methods (iTALS and iTALSx) from the point of view of item-to-item recommendations, and we propose how contextual information can be integrated into the similarity computation.

The paper is structured as follows. Section 2 briefly reviews the main advances in item-to-item recommendations and in context-awareness. The main idea of the paper (i.e. the incorporation of the context information into similarities) is presented in section 3. In the first part we describe the models we use; the second part deals with the proposed approach for each model. We report on our experiments and results in section 4, where the strengths and weaknesses of the different approaches are also discussed. Finally, section 5 concludes the paper.

2. RELATED WORK

Item-to-item recommendation — just like item-to-user recommendations — can be classified as content based filtering (CBF) or collaborative filtering methods (CF). Item-to-item CF methods are usually neighbor based [17], meaning that the similarities between items are defined as the similarities between the sets of transactions of the items. Another approach is to extract association rules [3] from the data. These rules then can be used to recommend items to the actual item (e.g. "who viewed this item, also viewed those items"). This method can perform well in certain application areas [9].

Context-aware recommender systems (CARS) [1] emerged as an important research topic in the last years. Recently, entire workshops were devoted to this topic on major conferences. The application fields of context-aware recommenders include among other movie [6] and music recommendation [5], point-of-interest recommendation (POI) [4], citation recommendation [11]. Context-aware recommender approaches can be classified into three main groups: pre-filtering, post-filtering and contextual modeling [2]. Baltrunas and Amatriain [5] proposed a pre-filtering approach by partitioned user profiles into *micro-profiles* based on the time split of user event falls, and experimented with different time partitioning. Post-filtering ignores the contextual data at recommendation generation, but disregards irrelevant items (in a given context) or adjust recommendation score (according to the context) when the recommendation list is prepared; see a comparison in [18]. Tensor factorization—falling into the contextual modeling category—was proposed as a solution for context-awareness in the factorization framework. There are different approaches for explicit [16, 19] and implicit feedback data [21, 14, 13].

3. CONCEPT

We present in this section how item-to-item recommendation can be performed using context-aware factorization models.

3.1 Context-aware factorization models

Before introducing the main idea of this paper, the base methods are to be introduced. Since we concentrate on the implicit feedback case, we review iALS, iTALS and iTALSx methods.

IALS [15] is the seminal method for handling implicit feedback in the factorization framework. It factorizes a full binary matrix — describing the user–item interactions — optimizing for weighted root mean squared error (wRMSE) using alternating least squares (ALS). A cell of the matrix is 1 only if there is at least one transaction between the user and the item. Therefore the number of zeroes is dominant over the number of ones in this matrix. The weight corresponding to a cell with zero is 1 and $\gg 1$ otherwise. The actual weight may also depend on the number of transactions between the given user and item. The method results in two low rank (feature) matrices: one for the users and one for the items. Each cell of the original matrix is approximated by the scalar product of the appropriate user and item feature vector. IALS computes the feature vectors efficiently by decomposing the derivatives at the minimizing step into user-dependent and user-independent parts (see details in [15]), thus the method scales linearly with the number of non-zeroes in the original matrix.

ITALS [14] is the generalization of IALS for tensors, and thus provides a solution for context-aware recommendations in the implicit case. An additional context dimension is added to the user–item setting thus extending the matrix of transactions to a tensor[1] and a feature matrix for each dimension (user, item, context(s)) are provided as output. ITALS uses the same approximation framework as IALS (binary input, weighted RMSE as objective function, and ALS as optimizer), but approximates the cells by the sum of elements in the Hadamard or elementwise product of three vectors for the given user, item and context-state. Since the derivatives involve more factors, the decomposition that allows for efficient calculation time is somewhat more complicated as for IALS. It is shown in [14] that the complexity of this algorithm can theoretically be the same as that of the iALS.

ITALSx [13] applies the binary tensor approach with a different approximation model. The cells are approximated by the sum of three scalar products: between the user–item, user–context and item-context feature vectors (see also Ta-

[1] More than one context dimensions might be added but it is not advised to use too many of them since then the tensor becomes extremely sparse.

Table 1: Main properties of used algorithms

Method	Model		
iALS	$\hat{r}_{u,i} = (P_u)^T Q_i$		
iTALS	$\hat{r}_{u,i,c} = 1^T (P_u \circ Q_i \circ C_c)$		
iTALSx	$\hat{r}_{u,i,c} = (P_u)^T Q_i + (P_u)^T C_c + (Q_i)^T C_c$		
Method	**Implicit?**	**Context?**	**Common properties**
iALS	Yes	No	modeling: binary tensor
iTALS	Yes	Yes	learning: ALS
iTALSx	Yes	Yes	optimizing: wRMSE

ble 1). It is shown in [13] that the model has the same complexity as ITALS, that is, it scales linearly with the number of non-zero inputs in the tensor. The main difference in angle lies within that iTALS uses context-state based reweighting of the user–item relations, while iTALSx considers the effect of context separately for the user and the item dimensions (by projecting the relation into two dimensional subspaces), but meanwhile the singular context feature vector (per context-state) still ensures that context features are influenced by both users and items.

Table 1 summarizes the properties of the three approaches. P, Q, and C are the user, item, and context feature matrices, each column of a matrix is a feature vector. Users, items and context-states are indexed by u, i and c respectively, while the P_u, Q_i and C_c denote the appropriate column of the matrix. The $\hat{r}_{u,i}$ value is the predicted preference of the user on the item (given the current context for context-aware methods).

3.2 Two levels of context-aware item-to-item recommendations

Similarity between items must be defined in order to provide item-to-item recommendations. In the factorization framework it is the similarity between item feature vectors. Here we investigate two alternatives for similarity: cosine similarity and scalar product. The main difference between them is that scalar product favors feature vectors with higher norm. Usually the norm of the feature vector is higher for more popular items, therefore those are more often similar to other items. This might increase accuracy (as a many users likes popular items), but might also reduce the diversity and the coverage of recommendations.

It is interesting to note that similarity between item feature vectors is used, but the models are optimized for predicting the user–item relations. This is because the desired similarity between items is not supplied for the system. Also it is assumed that a good factorization method produces similar feature vectors to similar items.

Context-aware methods are able to achieve better results compared to context-unaware ones. This is mainly because (1) good context separates items and/or users that behave differently, thus makes it easier to learn the user–item relations more efficiently; (2) different recommendations can be provided in different situations given that context-states differ. Based on these observations, two levels of context-awareness can be incorporated into factorization based similarities. The first level is to use the standard similarity

$$S_{i,j}^{(\mathrm{L1})} = \frac{(P_i)^T P_j}{\sqrt{(P_i)^T P_i \, (P_j)^T P_j}} \tag{1}$$

i.e. cosine similarity between item features, while learning a context-aware method. If the expressive power of context information is large the context-aware method will learn more accurate item models than the context-unaware factorization. Therefore the similarities between the items will be more precise. This approach is only based on (1) and while a legitimate strategy, it does not exploit the full potential of context information. For example item-2-item recommendation lists created in this way will be static, that is, he same set of items is recommended to a given item in every context-state.

The second level of context-awareness of item similarities leverages the context information through the feature vector of the context-state. The actual similarity value is calculated differently for different models[2]. Equation 2 and equation 3 defines the second level context-aware similarity for iTALS and iTALSx respectively[3]. Both similarities are context dependent therefore should be indexed with the selected context-state as well (in addition to the indices of the two compared items).

$$S_{i,j,c}^{(\mathrm{L2,iTALS})} = \frac{1^T (P_i \circ C_c \circ P_j)}{\sqrt{(P_i)^T P_i \, (P_j)^T P_j}} \tag{2}$$

$$S_{i,j,c}^{(\mathrm{L2,iTALSx})} = \frac{(P_i)^T P_j}{\sqrt{(P_i)^T P_i \, (P_j)^T P_j}} + \frac{(P_i)^T C_c}{\sqrt{(P_i)^T P_i}} + \frac{(P_j)^T C_c}{\sqrt{(P_j)^T P_j}} \tag{3}$$

Let us investigate the differences between the two approaches in detail. In $S^{(\mathrm{L2,iTALS})}$ the context feature vector re-weights the scalar product of the two item feature vectors. The importance of a feature depends on the selected context-state. Some items that are similar in one context-state may be different in another. Consequently, the approach is very sensible for the proper context selection: while context dimensions of context-states that suit the problem well may result in a much increased accuracy (as the model is fully context-aware), if the context-states are not appropriately selected

[2] The calculation of similarity values does not depend on the loss function, nor on the learning method.

[3] The equations present the calculations using cosine similarity. For scalar product, the normalization with the length of the item features should be omitted.

for the problem or the context features are not learned well, the similarities will be completely uncorrect.

As shown in our ongoing research [13], the personalized recommendations generated by iTALS may also suffer from the sensitivity to the proper context selection but the effect is much smaller there, because the feature vectors are optimized using the same model as with the predictions are made.

On the other hand, $S^{(\mathrm{L2,iTALSx})}$ is a more stable similarity function approach that depends on the context feature in less extent. The similarity model has three terms: the first context-independent term is borrowed from Eq. (1), while the other two terms are context-dependent capturing how the given context-state favors the i^{th} and j^{th} items. Unlike the $S^{(\mathrm{L2,iTALS})}$ similarity, the context feature has no effect on every term of the similarity prediction, therefore $S^{(\mathrm{L1})}$ acts as a regularization term, even if the context dependent promotions/demotions go astray.

We opt for not normalizing the context features in equations (2), since for the iTALS model, the length of the context feature has no effect on the item ranking in a given context-state. On the other hand the normalization of the context vector at the iTALSx model yields a different model, coined as iTALSxN. The main difference between these models is that the effect of the item demotions and promotions are stronger with the non-normalized version.

4. EXPERIMENTS

We used four genuine implicit feedback data sets (LastFM 1K [7], TV1, TV2 [8] and Grocery) to evaluate our algorithm. The properties of the data sets are summarized in Table 2. The column "Multi" shows the average multiplicity of user–item pairs in the training events.[4] Chronological train–test splits were created. The length of the test period was selected to be at least one day depending also on the domain and the frequency of events. We used the artists as items in LastFM.

Our primary evaluation metric is *recall@20*. Recall is defined as the ratio of relevant recommended items and relevant items. An item is considered relevant for a user if there is an event in the test set with the given user and item. Recall does not take into account the position of an item on the recommendation list. We choose cutoff 20 for several reasons. First, the length of the recommendation list is limited as users are exposed to a few recommended items at a time, and their depends on the user interface. However, during a user session, several recommendation widgets can be shown, so user may see 20 recommendations during a visit. Mean Average Precision (MAP) was used as a secondary evaluation metric. MAP considers the order of the recommended items thus preferring methods that put the relevant items at the beginning of the recommendation list. We also used a cutoff value of 20 for MAP (denoted as *MAP@20*).

We also calculated *coverage*, a metric that quantifies the ability of the recommender system to explore the entire item catalog [12]. We adopted a version of catalog coverage [10, 20], that we call *perceived catalog coverage*, the ratio of actually recommended items compared to the number of items measured over the entire test period. Larger coverage means that users get more diverse recommendations in general, that is usually considered as a desired property of the recommender system [20]. Usually more accurate recommender algorithms tend to have lower coverage [22], since—by the nature of the concept—many users prefer popular items, therefore models being biased towards popular items achieve higher accuracy but lower coverage.

Seasonality was used as context information, because it relies solely on time stamp of the events that is available in all of the data sets. On seasonality we mean that we define a periodicity and divide it into smaller time intervals called *time bands*. For example, hours of a day can be time bands of a day, or the weekdays and the weekend can be time bands of a week. The context of an event is the time band in which it happened. The periodicity (or season) was a week for the Grocery data set and one day for the others, because users tend to do shopping once or twice a week, while activities related to music listening or movies watching show daily periodicity. The time bands were the days of the week for Grocery and four hour long time windows for the other databases.

4.1 Initial results

The first experiment revolves around the usefulness of the context-aware similarities. The results are showcased in Table 3. Since no item-2-item similarities are given, we perform the evaluation schema as described next. We assume that if a user interacts with two subsequent items then the second one can be considered as a meaningful recommendation to the first one, if the time difference between the two events is not very large. Thus recall and MAP values were measured as follows: the events in the test set were ordered by their time stamps. A *preceding event* of an event is defined as the closest event within 24 hours. If there was no preceding event than no recommendations can be generated and the algorithms can not score on given test event, but it is still considered in the measurements. In case the an existing preceding event, an item-to-item recommendation list was generated to the item of the preceding event.

General observations: As it was expected, scalar product similarities generally achieve higher recall (and MAP) and lower coverage, because they prefer more popular items. The iTALSx-based model acts more similarly to the context-unaware similarity than the iTALS-based model. It is especially true when L2 similarities are used. The iTALS-models with L2 context-aware similarities seem to be very sensitive to the quality of the context dimension (e.g. performance on TV2 with season and on Grocery with seq. is very poor).

Grocery dataset: Recall is similar for the context-unaware and context-aware similarities using cosine similarity (both level 1 and 2). This implies that the average number of relevant items recommended are similar for each method. However the L2 similarity with the iTALS-based model achieves higher MAP. This means that the ranking of those items is better. The coverage of the recommendations is much higher for the context-aware methods using cosine similarity. With scalar product similarity the coverage also increases in most cases, the accuracy (recall and MAP values) of the algorithm is also significantly better, especially if iTALS-based models are used.

TV1 dataset: The results are similar with both cosine and scalar product similarity. The iTALS-based models perform similarly to the context-unaware method, while the iTALSx-based model performs slightly better. The only outstanding

[4] This value is 1.0 at two data sets as TV1 and TV2 data might have been filtered for duplicate events.

Table 2: Main properties of the data sets

Dataset	Domain	Training set				Test set	
		#Users	#Items	#Events	Multi	#Events	Length
Grocery	E-grocery	24947	16883	6238269	3.0279	56449	1 month
TV1	IPTV	70771	773	544947	1.0000	12296	1 week
TV2	IPTV	449684	3398	2528215	1.0000	21866	1 day
LastFM	Music	992	174091	18908597	21.2715	17941	1 day

Table 3: Results with context manually set

RECALL@20											
	Cosine similarity						Scalar product				
Dataset	Non-CA	CA Level1		CA Level2			Non-CA	CA Level1		CA Level2	
		iTALS	iTALSx	iTALS	iTALSx	iTALSxN		iTALS	iTALSx	iTALS	iTALSx
Grocery	0.0740	0.0738	0.0720	0.0729	0.0733	0.0836	0.0711	0.1323	0.0859	0.1244	0.0863
TV1	0.0587	0.0591	0.0621	0.0514	0.0617	0.0860	0.0548	0.0599	0.0629	0.0522	0.0843
TV2	0.2167	0.1915	0.2180	0.0183	0.2151	0.2097	0.1954	0.1515	0.1764	0.0364	0.1729
LastFM	0.1035	0.1098	0.0936	0.1051	0.1018	0.1002	0.0920	0.1073	0.0768	0.1099	0.0827
MAP@20											
	Cosine similarity						Scalar product				
Dataset	Non-CA	CA Level1		CA Level2			Non-CA	CA Level1		CA Level2	
		iTALS	iTALSx	iTALS	iTALSx	iTALSxN		iTALS	iTALSx	iTALS	iTALSx
Grocery	0.1237	0.1260	0.1123	0.2007	0.1244	0.1811	0.1326	0.4038	0.1677	0.4041	0.1689
TV1	0.0147	0.0160	0.0149	0.0142	0.0167	0.0294	0.0142	0.0172	0.0151	0.0144	0.0242
TV2	0.0657	0.0571	0.0638	0.0028	0.0623	0.0740	0.0576	0.0442	0.0489	0.0077	0.0494
LastFM	0.4883	0.5096	0.4226	0.6325	0.5020	0.5416	0.4571	0.5410	0.3670	0.6424	0.3459
COVERAGE											
	Cosine similarity						Scalar product				
Dataset	Non-CA	CA Level1		CA Level2			Non-CA	CA Level1		CA Level2	
		iTALS	iTALSx	iTALS	iTALSx	iTALSxN		iTALS	iTALSx	iTALS	iTALSx
Grocery	88.83%	94.06%	94.87%	93.40%	94.01%	89.61%	61.75%	63.81%	78.99%	85.44%	46.32%
TV1	87.84%	81.24%	89.13%	93.79%	93.01%	92.11%	79.82%	57.70%	79.30%	79.43%	75.29%
TV2	93.41%	93.64%	94.14%	58.62%	85.82%	83.70%	40.32%	19.48%	28.08%	20.66%	26.72%
LastFM	17.78%	18.57%	20.98%	9.02%	14.53%	13.87%	10.94%	11.36%	11.28%	3.08%	9.88%

result is the L2 iTALSx-based similarity with scalar product. Also, the coverage values for the L2 methods and cosine similarity outperform that of the basic method.

TV2 dataset: In earlier works we found that seasonality (with these settings at least) does not suit this data set as a context dimension. Thus it is surprising that the context-aware approaches are on par with the basic approach. The iTALS-based models are slightly worse, while iTALSx-models achieve approximately the same results when cosine similarity is used. ITALS L2 similarity has poor performance with scalar product, because this approach is very sensitive to the quality of context.

LastFM dataset: The performance of iTALS-based methods are a little bit better, while iTALSx-based approaches are slightly outperformed by the basic method. However, there is a notable increment in the coverage for context-aware methods, except for L2 iTALS-based.

4.2 Sensitivity to context quality

The second experiment examines the sensitivity of the approaches to context quality (see Table 4). Here the context-states were created automatically using a simple clustering within the context dimension to suit the data better [13]. As shown in [13], the automatically determined context-states performs usually better than the manually constructed ones.

The improved quality of the context dimension enables the context-aware methods to perform better. Apparently the iTALS-based approach can benefit more from this than the other model, because it is more sensitive. The usage of automatic context selection improved the results by an average of 1.62% and 4.01% for iTALS and iTALSx models on Level1 and by 216.35% and 11.06% respectively on Level2. We highlight two extreme cases from the results. The first is context-aware, iTALS-based L2 similarity on TV1. Here recall is 7–9 times the recall of the basic approach. The actual ratio depends on the similarity metric. The other one is the same method on TV2. In this case the recall is a small fraction of that of the basic method. These examples demonstrates how the similarity metric L2 with ITALS is sensitive to the proper context values.

4.3 Which approach to use?

The experiments show that each method has its strengths and weaknesses, thus a clear winner can not be declared. If the quality of the context dimension (w.r.t. the problem) is acceptable than L1 methods with either model can be used to increase coverage without loss in accuracy. The relative performance of iTALS and iTALSx models depends on the

Table 4: Results with context automatically set

RECALL@20											
	Cosine similarity						Scalar product				
Dataset	Non-CA	CA Level1		CA Level2			Non-CA	CA Level1		CA Level2	
		iTALS	iTALSx	iTALS	iTALSx	iTALSxN		iTALS	iTALSx	iTALS	iTALSx
Grocery	0.0740	0.0769	0.0766	0.0899	0.0958	0.1021	0.0711	0.1341	0.0956	0.1373	0.1013
TV1	0.0587	0.0586	0.0636	0.5270	0.0630	0.0796	0.0548	0.0601	0.0660	0.3958	0.1108
TV2	0.0657	0.1992	0.2175	0.0383	0.2162	0.2084	0.1954	0.1534	0.1747	0.0309	0.1735
LastFM	0.1035	0.1090	0.0953	0.1239	0.1011	0.1044	0.0920	0.1108	0.0817	0.1116	0.0881

MAP@20											
	Cosine similarity						Scalar product				
Dataset	Non-CA	CA Level1		CA Level2			Non-CA	CA Level1		CA Level2	
		iTALS	iTALSx	iTALS	iTALSx	iTALSxN		iTALS	iTALSx	iTALS	iTALSx
Grocery	0.1237	0.1386	0.1294	0.2040	0.2737	0.3299	0.1326	0.3990	0.2350	0.4327	0.2790
TV1	0.0147	0.0158	0.0156	0.1071	0.0165	0.0293	0.0142	0.0166	0.0160	0.0929	0.0466
TV2	0.0657	0.0592	0.0613	0.0118	0.0631	0.0771	0.0576	0.0445	0.0473	0.0073	0.0494
LastFM	0.4883	0.5036	0.4275	0.8078	0.4792	0.4849	0.4571	0.5572	0.3718	0.6870	0.3847

used similarity metric and the properties of the data set. For example: data with stronger popularity-effect can tolerate more noise in the context dimension when scalar product similarity is used instead of cosine similarity. Generally, the iTALS-based model should be applied in such a setting that tolerates better the noise in the context.

If the context quality is fair, then L2 similarities can greatly improve performance. The iTALS model is often better in this scenario, and may increase recall beyond expectations (see TV1 results in Table 4). However the sensitivity of iTALS to context selection is a double-edged sword, and thus can yield in large performance decrease (see TV2 results in Table 4). If the context quality is poor, then no improvement can be expected from context-aware methods in general, including context-aware similarities as well. In such a case, one should stick to be original context-unaware approach.

5. CONCLUSION

This paper proposed context-aware similarity functions based on the feature vectors of context-aware factorization methods. Two levels of context-awareness were defined. On the first level, context is only used to enable the algorithm to be able to learn the item features better, but the similarities themselves are not context-aware. The more advanced level-2 approaches define context-dependent similarity functions.

We measured recall, MAP and coverage in the experiments performed on four implicit feedback data sets, but not clear winner could be specified. The proposed approaches can improve either accuracy (recall and MAP) or coverage, or in certain cases both. The context independent level-1 methods, where context is only used during the learning, perform consistently, but usually only slight improvement can be achieved over the baseline. On the other hand, context dependent similarity approaches are very sensitive to the context quality. This can result both in outstanding and in very poor performance. The sensitivity is different for the two investigated models: the Hadamard product based iTALS is much more sensitive than the pairwise product based iTALSx model.

The results suggest that in a practical application scenario, context-aware item-2-item recommendation algorithms are advised to be used since at least a small improvement can always be achieved with level-1 approaches. From level-2 approaches, the more robust iTALSx-based method can be used safely almost every time, while on the other hand, the more sensitive level-2 iTALS-based methods should be used with care, but can greatly improve the performance if used with proper context dimension.

Future work includes the through examination of context dimensions and context states that will enable us to predict which context dimension will prove useful for a given problem and with a given model. We assume that some context dimensions are more suitable for creating context-aware item biases (or in other words: too promote and demote items), while others are more useful in a reweighting setting, and we assume that this property of the context can be predetermined from the data.

6. REFERENCES

[1] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst. 23*, 1 (2005), 103–145.

[2] Adomavicius, G., and Tuzhilin, A. Context-aware recommender systems. In *Recsys'08: ACM Conf. on Recommender Systems* (Lausanne, Switzerland, 2008), 335–336.

[3] Agrawal, R., Imieliński, T., and Swami, A. Mining association rules between sets of items in large databases. In *SIGMOD'93: ACM SIGMOD Int. Conf. on Management of Data* (Washington DC, USA, 1993), 207–216.

[4] Bader, R., Neufeld, E., Woerndl, W., and Prinz, V. Context-aware POI recommendations in an automotive scenario using multi-criteria decision making methods. In *CaRR'11: Workshop on Context-awareness in Retrieval and Recommendation* (Palo Alto, CA, USA, 2011), 23–30.

[5] Baltrunas, L., and Amatriain, X. Towards time-dependant recommendation based on implicit

feedback. In *CARS'09: Workshop on Context-aware Recommender Systems* (New York, NY, USA, 2009), 1–5.

[6] Bogers, T. Movie recommendation using random walks over the contextual graph. In *CARS'10: 2nd Workshop on Context-Aware Recommender Systems* (Barcelona, Spain, 2010), 1–5.

[7] Celma, O. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[8] Cremonesi, P., and Turrin., R. Analysis of cold-start recommendations in iptv systems. In *Proceedings of the 2009 ACM Conference on Recommender Systems* (2009).

[9] Davidson, J., Liebald, B., Liu, J., et al. The YouTube video recommendation system. In *Recsys'10: ACM Conf. on Recommender Systems* (Barcelona, Spain, 2010), 293–296.

[10] Ge, M., Delgado-Battenfeld, C., and Jannach, D. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Recsys'10: 4th ACM Conf. on Recommender Systems*, ACM (New York, NY, USA, 2010), 257–260.

[11] He, Q., Pei, J., Kifer, D., Mitra, P., and Giles, L. Context-aware citation recommendation. In *WWW'10: Int. Conf. on World Wide Web* (Raleigh, NC, USA, 2010), 421–430.

[12] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22*, 1 (2004), 5–53.

[13] Hidasi, B. Technical report on iTALSx. Tech. Report Series 2012-2, Gravity R&D Inc., 2012.

[14] Hidasi, B., and Tikk, D. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'12)* (Bristol, UK, September 24–28, 2012).

[15] Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *ICDM-08: IEEE Int. Conf. on Data Mining* (Pisa, Italy, 2008), 263–272.

[16] Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Recsys'10: ACM Conf. on Recommender Systems* (Barcelona, Spain, 2010), 79–86.

[17] Linden, G., Smith, B., and York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE 7*, 1 (2003), 76–80.

[18] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Recsys'09: ACM Conf. on Recommender Systems* (New York, NY, USA, 2009), 265–268.

[19] Rendle, S., and Schmidt-Thieme, L. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10 (2010), 81–90.

[20] Shani, G., and Gunawardana, A. Evaluating recommendation systems. In *Recommender System Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, Berlin–Heidelberg, 2011, 257–297.

[21] Wermser, H., Rettinger, A., and Tresp, V. Modeling and learning context-aware recommendation scenarios using tensor decomposition. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, IEEE (2011), 137–144.

[22] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. Solving the apparent diversity–accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences 107*, 10 (2010), 4511–4515.