Using Contextual Information as Virtual Items on Top-N Recommender Systems

Marcos A. Domingues
Fac. of Science, U. Porto
LIAAD-INESC Porto L.A.
Portugal
marcos@liaad.up.pt

Alípio Mário Jorge
Fac. of Science, U. Porto
LIAAD-INESC Porto L.A.
Portugal
amjorge@fc.up.pt

Carlos Soares
Fac. of Economics, U. Porto
LIAAD-INESC Porto L.A.
Portugal
csoares@fep.up.pt

ABSTRACT

Traditionally, recommender systems for the Web deal with applications that have two dimensions, users and items. Based on access logs that relate these dimensions, a recommendation model can be built and used to identify a set of N items that will be of interest to a certain user. In this paper we propose a method to complement the information in the access logs with contextual information without changing the recommendation algorithm. The method consists in representing context as virtual items. We empirically test this method with two top-N recommender systems, an item-based collaborative filtering technique and association rules, on three data sets. The results show that our method is able to take advantage of the context (new dimensions) when it is informative.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Induction*

General Terms

Measurement, Experimentation, Algorithms

Keywords

Context, Multidimensional data, Recommender system

1. INTRODUCTION

Most Web sites offer a large number of information resources to their users. Finding relevant content has, thus, become a challenge for users. Recommender systems have emerged in response to this problem. A recommender system for a Web site receives (implicit or explicit) information about users and their behavior and recommends items that are likely to fit his/her needs [12].

Recommender models for Web personalization can be built from the historical record of accesses to a site, where one access is a pair $< user\_id, item >$. Each access is interpreted as a rating of 1 given by the user to the item. However, other dimensions, such as time and location, can add contextual information and improve the accuracy of recommendations. For instance, the type of books that a user looks for in Amazon during work hours is probably different from the books searched for during leisure hours.

According to [11], the idea that contextual information is important when predicting customer behavior is not new. Many Web sites are supported by Content Management Systems (CMS), that often store much contextual information. However, this is not true in all cases and, additionally, getting information that is really relevant for recommendation is a hard task in many applications [8]. Adomavicius et al. [1] have investigated the use of context for rating estimation in multidimensional recommender systems. Palmisano et al. [11] have used contextual information to improve the predictive modeling of customer's behavior. Both authors have developed a special-purpose browser to obtain rich contextual information.

In this paper we exploit how contextual information can be used to improve the accuracy of Top-N Recommender Systems. Existing contextual recommender systems typically use contextual information as a label for segmenting/filtering sessions, using them to build the recommendation model (e.g., [1, 11]). We follow an alternative approach, which uses the contextual attribute as a virtual item. This means that it is treated as an ordinary item for building the recommendation model, which has the advantage of allowing the use of existing recommendation algorithms. As our contextual information are obtained from multidimensional data, we have called our approach **DaVI** (*Dimensions as Virtual Items*). Instead of a special-purpose browser [1, 11], we collect the multidimensional data from Web access logs and from attributes stored in databases of the Web sites. We have empirically tested our approach with two recommendation techniques, item-based collaborative filtering and association rules, to assess the effect of adding context on the accuracy of traditional Web recommender systems. We present results obtained on three data sets.

In the following section, we present the contextual information used in our experiments. Next, we describe the recommendation techniques and the approach proposed. Then, we discuss results and present conclusions and future work.

2. CONTEXTUAL INFORMATION

There are many definitions of context in the literature depending on the field of application and the available customer data [11]. In this paper, context is defined as any

CARS-2009, October 25, 2009, New York, NY, USA.

information that can be used to characterize the situation of an entity [5]. Here an entity is an access to an item/Web page by a user.

A critical issue is how to obtain the rich contextual information [6]. In some circumstances, context is explicit, such as a person informing a movie recommender system where he/she wants to watch a movie. On the other hand, the contextual information can also be inferred from Web access data. For example, we can observe if a person bought an item, from an e-commerce Web site, on a weekday or a weekend, from the Web access logs.

Besides general contextual information that can be obtained from access logs, we may use domain-specific information, that is typically collected from the CMS. For example, if an item represents an access to a music, the genre of the music can be used as a dimension of contextual information.

In Table 1 we present the dimensions/contextual information considered in the experiments presented in this paper. The first group of contextual information was obtained by pre-processing Web access logs. The second group was collected from the CMS of a Web site of Portuguese Music[1] used in this study. The last group refers to a public data set[2] that contains a record of user interactions with the Entree Chicago restaurant recommender system. All the information is stored in a data warehouse that was specifically designed for modeling Web sites [7].

Table 1: Contextual information

Context	Description
day	Day of each access (from 01 to 31).
month	Month of each access (from 01 to 12).
week_day	Week day of each access (from Monday to Sunday).
work_day	If the accesses were made during the week (from Monday to Friday) or weekend (Saturday or Sunday).
hour	Hour of each access (from 01 to 24).
work_hour	If the accesses were made during the working time (from 8 a.m. to 6 p.m.) or not.
location	Location where the accesses were made (country).
music_genre	The genre of a music. There are 45 different musical genres, for instance, pop, rock, jazz, and so forth.
band	The band which plays a music. There are 2296 different bands in our music recommendation data sets.
instrumental	If a music is instrumental or not.
intention	The intention of navigation in a restaurant recommendation system (for example, the search for a restaurant cheaper, closer, more traditional, more creative, and so forth). There are 9 different intentions of navigation in our experiments.

3. RECOMMENDER SYSTEMS

A recommender system for the Web typically outputs an ordered list of recommendations, given a trail of recent Web page requests. Historical information about the behavior of the users of the site and the current session are used to suggest certain pages or services, or even the purchase of certain products [12]. In the context of the Web, a session can be abstracted to a set of pairs $< user\_id, item >$, recorded at moments close in time, with the same *user_id*.

Usually a recommender system is divided into a two-stage process [3]. The first stage is carried out offline. Data representing the behavior of users of the Web site, which was previously collected are mined and a model is generated for use in future online interactions. The second stage is carried out in real-time with a new user interacting with the Web site. Data from the current user session are used as input by the model to generate a list of N recommendations. A number of algorithms have been used for offline model building [3], including Collaborative Filtering, Item-Based Collaborative Filtering, Association Rules and Markov Models. In this section we briefly describe the two algorithms used in this work, Item-Based Collaborative Filtering and Association Rules, and how we have applied **DaVI** (*Dimensions as Virtual Items*) on these algorithms.

3.1 Item-Based Collaborative Filtering

Item-based collaborative filtering (CF) analyzes stored accesses (grouped in sessions) to identify relations between the items in the set I, which contains all items of a Web site [10]. The recommendation model is a matrix representing the similarities between all pairs of items, according to a chosen similarity measure. An abstract representation of a similarity matrix is shown as Table 2.

Table 2: Item-item similarity matrix.

	i_1	i_2	$\cdots$	i_k
i_1	1	$sim(i_1, i_2)$	$\cdots$	$sim(i_1, i_k)$
i_2	$sim(i_2, i_1)$	1	$\cdots$	$sim(i_2, i_k)$
$\cdots$	$\cdots$	$\cdots$	1	$\cdots$
i_k	$sim(i_k, i_1)$	$sim(i_k, i_2)$	$\cdots$	1

In Table 2, each item $i \in I$ is an accessed page. The similarity measure used here is the cosine angle, defined by

$$sim(i_{k_1}, i_{k_2}) = cos(\overrightarrow{i_{k_1}}, \overrightarrow{i_{k_2}}) = \frac{\overrightarrow{i_{k_1}} . \overrightarrow{i_{k_2}}}{||\overrightarrow{i_{k_1}}|| * ||\overrightarrow{i_{k_2}}||},$$

where $\overrightarrow{i_{k_1}}$ and $\overrightarrow{i_{k_2}}$ are binary vectors with as many positions as existing users. The value 1 means that the users accessed the respective item/page. The value 0 is the opposite. The "." denotes the dot-product of the two vectors.

Given a user who accessed the set of items $O \subseteq I$, the model generates a recommendation by selecting the N which are the most similar to the items in the set O. Here, the similarity for each item $i \notin O$ is given by the weighted average of its nearest neighbors with respect to their presence in the set O.

Table 3: Similarity matrix with the contextual information *day*.

	i_1	$\cdots$	i_k	$\cdots$
i_1	1	$\cdots$	$sim(i_1, i_k)$	$\cdots$
$\cdots$	$\cdots$	1	$\cdots$	$\cdots$
i_k	$sim(i_k, i_1)$	$\cdots$	1	$\cdots$
d_1	$sim(d_1, i_1)$	$\cdots$	$sim(d_1, i_k)$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
d_v	$sim(d_v, i_1)$	$\cdots$	$sim(d_v, i_k)$	$\cdots$

When we apply **DaVI** on the item-based collaborative filtering algorithm, it treats the contextual attributes as new

[1] http://www.palcoprincipal.pt.

[2] http://archive.ics.uci.edu/ml/datasets/Entree+Chicago+Recommendation+Data.

items (virtual items) in the data set. This means that it adds a new row and column for each different value of the context to the former similarity matrix and calculates the corresponding similarity values, among the values of the context and the other items, as presented previously. A representation of a similarity matrix with contextual information $day = \{d_1, d_2, \cdots, d_v\}$ is shown on Table 3. Here, an item can be a page or a possible value for the context day (1 to 31). Although the contextual information is used in the models, only pages are recommended. The recommendations will be the set of pages that are most similar to a given set of observable items $O \subseteq \{I \cup day\}$. The rationale behind this approach is that the similarity between a given item and a given day (for example) is higher if the item tends to be accessed on that day of the month. This way, the relation between items and the context is captured. When a recommendation is made for an active session, the value of the context on that particular session (e.g., the day of the month the active session is taking place) is used to provide the contextual information.

3.2 Based on Association Rules

A recommendation model M based on association rules (AR) is a set of rules R, each of them with the form $A \rightarrow B$, where A and B are sets of items. Each AR is characterized by their support and confidence [2]. The model is generated from a set of Web sessions, consisting of a set of pairs $< id, item >$ with the same id, where id and $item$ identify the user and the accessed page. Given a set of observable items O, the set of rules R is used to recommend a set of items/pages $Recs$, as follows:

$$Recs = \{consequent(r_i) | r_i \in M \text{ and } antecedent(r_i) \subseteq O \\ \text{and } consequent(r_i) \notin O\}.$$

To obtain the top N recommendations, we select from $Recs$ the distinct recommendations corresponding to the rules with the highest confidence. In our work we use the ***Caren***[3] association rules generator.

Extending AR to handle contextual information by applying **DaVI**, simply consists of including extra pairs user-item into the former set of sessions. For example, to use the dimension *day*, we add a pair $< id, day = value >$ to the respective session with tag id, where $day = value$ represents the day of the month when the session id occurred. The set of augmented sessions are used as input to the recommendation algorithms. The rules built will include both actual items and virtual items on the antecedent and only actual items on the consequent. Given an active session occurring on day x, the set of observables O includes the items in the active session and the virtual items (e.g. $day = x$).

Notice that **DaVI** does not modify the recommendation algorithms. It just inserts the contextual information as virtual items in the data sets. Thus we can easily extend **DaVI** to other recommendation methods.

4. EMPIRICAL EVALUATION

In this section we evaluate how **DaVI** can improve the accuracy of the recommendation algorithms presented in Sections 3.1 and 3.2.

4.1 Experimental Setup

The evaluation is carried out on three different data sets (Table 4). The *Listener* data set contains accesses to songs in the music Web site mentioned earlier. The *Playlist* data set represents the set of songs explicitly selected by users of the same site for their individual playlists. *Entree* is a public data set that contains a record of user interactions with the Entree Chicago restaurant recommender system.

Table 4: Characteristics of the data sets

Data sets	# Accesses	≠ Items	≠ Users
Listener	62208	6428	9740
Playlist	37022	5428	4417
Entree	149849	639	31440

To measure the accuracy of the recommender systems we use the All But One protocol [4]. In this protocol, the sessions in the data set are split randomly into train and test. In our case, 80% for training and 20% for testing. The training set is used to generate the recommendation model (similarity matrix or association rules). For each session in the test set we randomly delete one pair $< id, item >$, referred to as *hidden* item. The remaining pairs represent the set of observables, O, based on which the recommendation is made.

The model is evaluated by comparing, for each session in the test set, the set of recommendations it makes (Rec), given the set of observables, O, against the hidden item. The set of recommendations $rec_1, rec_2, ..., rec_N$ for a given user id is represented as $\{< id, rec_1 >, < id, rec_2 >, ..., < id, rec_N >\}$ and N is the number of recommendations produced by the model. Based on the set of recommendations and the hidden item for all the session in the test set, we measure Recall, Precision and the F1 metric [13, 12]:

$$Recall = \frac{|Hidden \cap Rec|}{|Hidden|},\ Precision = \frac{|Hidden \cap Rec|}{|Rec|},$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}.$$

Recall corresponds to the proportion of relevant recommendations. Precision gives us the quality of each individual recommendation. F1 is a measure that combines Recall and Precision with an equal weight. It ranges from 0 to 1 and higher values indicate better recommendations. Global recall, precision and F1 are obtained by averaging individual test user values.

For the recommendation models based on association rules (AR), we chose a minimum support value trying to keep at least 50% of the items of the data sets for building the models. The minimum confidence values were defined as being the support value of the third most frequent item.

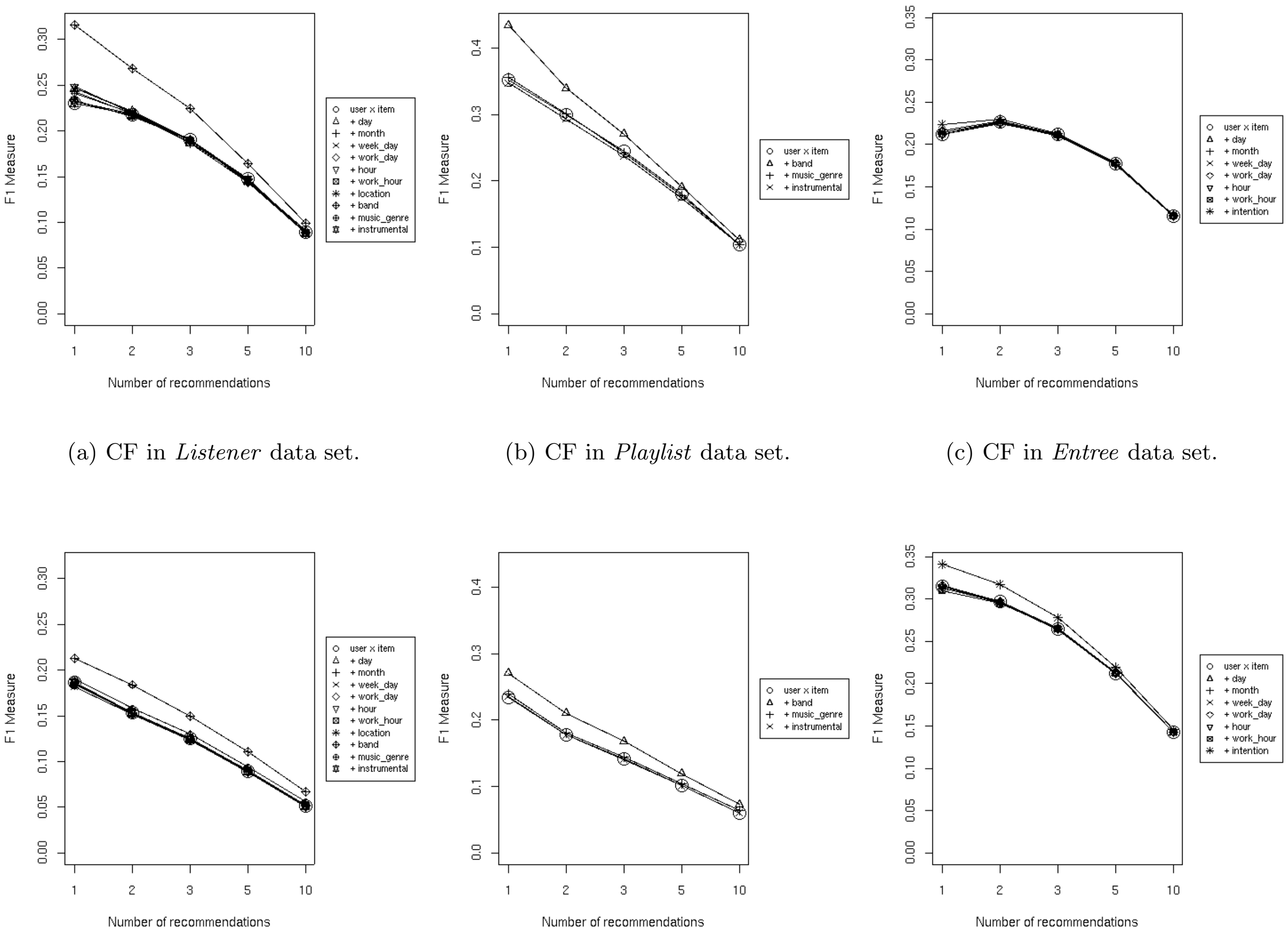
4.2 Single Dimension

Here we compare the results of the two algorithms using the traditional model (user × item) and with **DaVI**, applied separately to each contextual dimension presented in Table 1. The charts in Figure 1 plot the F1 measure.

4.2.1 Item-Based Collaborative Filtering

Our results show that **DaVI** improves item-based CF predictive performance when there is a rich contextual dimension. We can observe this with the dimension *band* in *Listener* and *Playlist* data sets (Figure 1 (a) and (b)). In Figure 1 (a) the dimension *band* yields a maximal value of 0.31

[3] http://www.di.uminho.pt/~pja/class/caren.html.



(a) CF in *Listener* data set.

(b) CF in *Playlist* data set.

(c) CF in *Entree* data set.

(d) AR in *Listener* data set.

(e) AR in *Playlist* data set.

(f) AR in *Entree* data set.

Figure 1: F1 metric for *Listener*, *Playlist* and *Entree* data sets.

(top 1). This value represents an F1 average gain of 34% compared to the value of F1 without applying **DaVI**. In Figure 1 (b) *band* provides a maximal value of 0.43 (top 1). This value represents a gain of 24%. In *Entree* data set (Figure 1 (c)), the only expressive gain using contextual information was obtained with the dimension *intention*, F1 value of 0.22 (top 1). This value represents a small gain of 5%, which means that the CF model worked worse with *intention* than *band*, used in the former data sets. Additionally, we can see that the highest gains were obtained using dimensions collected from databases of the Web sites. The gains using dimensions pre-processed from Web access logs were very small. The values show average gains around 1.3%, which means that the context inferred from Web access logs is not so rich in information.

4.2.2 Association Rules

Considering the association rules (AR) technique, our results also show that **DaVI** improves the accuracy of the recommendation models. In Figure 1 (d), we have a maximum value of 0.21 (top 1 with *band*). This value represents a gain of 14.5% compared to the value of F1 without any contextual information. In Figure 1 (e), the F1 measure for the dimension *band* and the top 2 has the maximal F1 gain, 23.5%. With respect to the *Entree* data set (Figure 1 (f)) we have a maximal value of 0.34 (top 1 with *intention*). This value represents a gain of 9.6%. An interesting fact here is that contrarily to the other data sets, the *Entree* presented highest F1 values with the association rules technique then with the item-based collaborative filtering.

4.3 Multiple Dimensions

So far, we applied **DaVI** to one contextual dimension at a time. However, it may be applied to several dimensions. We consider two different scenarios. The first one (called *all together*) simply applies the method to all dimensions presented in Table 1 simultaneously. The second scenario (called *forward selection*) uses a sequential forward selection algorithm [9], on the training data set, to select the best combination of dimensions that will be used to make recommendations. The algorithm starts from an empty set and sequentially adds the dimension d that results in the highest objective function $F(D + d)$ on a validation data set, when combined with the dimensions D that have already been selected.

Given that there are no other methods that combine several contextual dimensions for Top-N recommendation, we compared our method to an adaptation of the Combined Reduction approach [1] for this task. Briefly, this approach uses the values of the context/dimension as labels for segmenting Web accesses and was originally developed for rating estimation. It consists of the following two phases. First, using the training data, a recommendation method is run for each contextual segment (e.g. accesses on Mondays would be a segment) to determine which ones outperform the traditional model (using only user-item information). Second, taking into account the context of the active session, we choose the best contextual model to make the recommendation. Here the best model is the one which has the highest F1 value. Here, we have adapted it for the Top-N recommender algorithms presented in Sections 3.1 and 3.2.

As baselines, we have used the traditional user-item approach and also the results of the best individual dimension (called *best context*), according to the previous experiments.

In Table 5, the results for $N = 1$ show us that **DaVI**, using the best dimension, has F1 values equal or higher than other **DaVI** scenarios. The only exception is the *Listener* data set with the CF model, where the best is **DaVI** using all dimensions. The Combined Reduction approach has values equal to and better than **DaVI** (best context), respectively, in *Listener* and *Playlist* data sets with the AR model. In Table 5, the symbol “-” means that the algorithm timed-out.

Table 5: F1 measure for Top-1 recommendations

Methods	*Listener*	CF *Playlist*	*Entree*
user × item	0.230	0.351	0.211
DaVI (best context)	0.315	**0.434**	**0.225**
DaVI (forward selection)	0.311	0.416	0.210
DaVI (all together)	**0.317**	0.429	**0.225**
Combined Reduction	0.225	0.351	0.212
		AR	
user × item	0.186	0.234	0.315
DaVI (best context)	**0.213**	0.270	**0.341**
DaVI (forward selection)	0.203	0.261	**0.341**
DaVI (all together)	-	0.268	0.336
Combined Reduction	**0.213**	**0.280**	0.309

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a direct approach, called **DaVI**, that enables existing recommender systems to take advantage of contextual information as virtual items. We discussed the results obtained using two recommendation techniques, item-based collaborative filtering and association rules. Using **DaVI** with rich contextual information has revealed a great potential to improve the accuracy of recommender systems. However identifying rich contextual dimensions is not an easy task.

We have also compared different settings using the **DaVI** approach (best dimension, forward selection and all dimensions) with a more sophisticated Combined Reduction approach. Next, we will improve this empirical study and propose a method to identify rich contextual information from Web sites that can be used with **DaVI**.

6. ACKNOWLEDGMENTS

Fund. Ciência e Tecnologia (PhD Grant SFRH/BD/22516 /2005) and grant QREN-AdI Palco3.0/3121 PONORTE.

7. REFERENCES

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

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of Twentieth International Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] S. S. Anand and B. Mobasher. Intelligent techniques for web personalization. In *Intelligent Techniques for Web Personalization (ITWP 2003), LNCS 3169*, pages 1–36, 2003.

[4] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[5] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[6] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2):97–166, 2001.

[7] M. A. Domingues, A. M. Jorge, C. Soares, J. P. Leal, and P. Machado. A data warehouse for web intelligence. In *Proceedings of the 13th Portuguese Conf. on Artificial Intelligence*, pages 487–499, 2007.

[8] M. Gorgoglione, C. Palmisano, and A. Tuzhilin. Personalization in context: Does context matter when building personalized customer models? In *ICDM ’06: Proceedings of the Sixth International Conference on Data Mining*, pages 222–231, 2006.

[9] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, Feb 1997.

[10] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM’01: Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 247–254, 2001.

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

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conf. on Electronic Commerce*, pages 158–167, 2000.

[13] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’99)*, pages 42–49, 1999.