Modeling Short-Term Preferences in Time-Aware Recommender Systems

Pierpaolo Basile, Annalina Caputo, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro

Department of Computer Science, University of Bari Aldo Moro,
Via E. Orabona 4, 70125 Bari, Italy
{name.surname}@uniba.it
http://www.di.uniba.it

Abstract. Recommender Systems suggest items that are likely to be the most interesting for users, based on the feedback, i.e. ratings, they provided on items already experienced in the past. Time-aware Recommender Systems (TARS) focus on temporal context of ratings in order to track the evolution of user preferences and to adapt suggestions accordingly. In fact, some people's interests tend to persist for a long time, while others change more quickly, because they might be related to volatile information needs. In this paper, we focus on the problem of building an effective profile for short-term preferences. A simple approach is to learn the short-term model from the most recent ratings, discarding older data. It is based on the assumption that the more recent the data is, the more it contributes to find items the user will shortly be interested in. We propose an improvement of this classical model, which tracks the evolution of user interests by exploiting the content of the items, besides time information on ratings. When a new item-rating pair comes, the replacement of an older one is performed by taking into account both a decay function for user interests and content similarity between items, computed by distributional semantics models. Experimental results confirm the effectiveness of the proposed approach.

Keywords: Time-aware Recommender Systems, Content-based Filtering, Short-Term Preferences, Distributional Semantic Models

1 Introduction

Recommender systems adopts information filtering algorithms to suggest items or information that might be interesting to users. In general, these systems analyze the past behavior of a user, build a model or profile of her interests, and exploit that profile to find potentially interesting items. In collaborative approaches, the user profile is usually the vector of ratings assigned to *all* items that they have accessed, viewed, or purchased [12]. Content-based approaches rely on item and user descriptions (content) to build item representations and user profiles that suggest items similar to those a target user already rated (and liked) in the past [17].

One limitation of these traditional approaches is that the temporal context of ratings is not taken in account, but actually user preferences are likely to change over time: long term interests stay stable for a long time, short term preferences tend to vary with higher frequency. There are some domains, such as news recommendation, in which retaining this temporal distinction among user preferences has an impact on the accuracy of suggestions [7, 4]. Indeed, the issue of including time information into user modeling and recommendation approaches has been investigated early in literature [3, 22], but the topic recently received renewed attention, due to the significant improvements of recommendation accuracy obtained by the time-aware algorithm adopted by the winning team of Netflix Prize competition [15].

Time-aware Recommender Systems (TARS) fall in the more general category of contex-aware ones, that exploit the context in which users express their preferences (such as: location, *time*, weather, emotional state) in order to adapt rating prediction depending on the situation in which they are experiencing an item. TARS focus on temporal context of ratings to adapt the recommendation list accordingly. Regarding the usage of time information, the literature distinguishes two classes of methods [6]:

- *time-aware* approaches, that adapt rating predictions on the target recommendation time;
- time-adaptive approaches, which do not differentiate rating predictions according to the target time, but rather adjust some parameters or data dynamically.

We focus on time-adaptive approaches, specifically on those adopting some heuristics to penalize older preferences that are presumed to be less valid at recommendation time. These methods could be considered as a particular case of time decay heuristics, but they do not target a specific recommendation time (morning, week-end, etc.). Our investigation specifically concerns approaches that adopt some time-based strategy to learn separate models for short-term preferences and for interests that persists for a long time.

In particular, in this paper, we face with the problem of building an effective short-term preference model able to predict items that will shortly be consumed by the user. To address this issue, a simple and popular approach is the use of sliding windows, that learns the model by including in the training set only the most recent ratings, while older data are discarded or weighted so that they contribute to the model in a limited way. The literature reports controversial results about the adoption of time weights for ratings provided at different times. For example, in [10] the authors show that recommendation accuracy is improved by an exponential decay function, while an opposite conclusion is drawn by performing rating prediction on the Netflix Prize dataset [14].

We argue that recent ratings of users reflect their short-term preferences more than old ratings, but this is not true for *all* old ratings, but only for *those provided on items which are different from the recently rated ones.* In other words, the hypothesis is that older ratings do not correspond definitely to older

preferences, as assumed in the classical sliding window approach, but content of the items should be taken into account as well, in order to discard old interests which differ from new ones.

The research question we want to investigate in this work is: "Is the gradual decay of influence of ratings, combined with content similarity between items, useful for modeling short-term preferences?"

We propose a time-aware distributional content-based recommender system which suggests items the user will shortly be interested in. It exploits both an exponential decay function and semantic similarity between item descriptions for regulating the item participation in building of the user profile. Items are described in a WordSpace through the geometrical metaphora of meanings. Related words are represented as near points (i.e. vectors), while the semantics of item descriptions (i.e. text fragments) is computed by summing the vectors associated with their words.

The paper is organized as follows: in Section 2 we discuss some relevant literature and compare existing approaches to the proposed one, described in the following section. Section 4 analyzes the results of the experiments performed to validate our proposal, while conclusions are drawn in the last section.

2 Related Work

Following the characterization given by [6], we would place our method in the class of *time-adaptive heuristic-based approaches*. Rather than modeling the notion of time as a context with respect to the items may be relevant or not, we consider time as a continuos context attribute with respect to items could be either fresh or not. Simply stated, our time-context definition aims at modelling the recency of user's preferences. This point of view has been inspired by early works by Ding et al. [10, 11], that aimed at adapting collaborative filtering algorithms in order to capture preference drift.

In [10], the authors propose a novel item-based algorithm (that identifies the similarity between two items by comparing users' ratings on them), enhanced by a time-decay function in a way that the items rated recently contribute more to the prediction of the recommendations. The underlying assumption is the same as in the sliding window approach: latest ratings reveal latest interests. The proposed approach showed the actual improvement of precision obtained by using an exponential decay function to weight ratings. In the later work [11], a recency-based approach is proposed, in which each rating of the target user is assigned a weight that is computed according to its deviation from her most recent ratings on similar items. We ground our approach on similar basis, but a significant difference is that we adopt a decay function to weigh content similarity among items recently rated and older ones, in order to select those to be included in the training set for our model. Time-adaptive heuristics have been used also by Cao [8] et al. and Lathia et al.[16]. The former approach introduces four types of user interest patterns and proposes an effective approach for detecting these patterns by exploiting user rating graphs and rating chains. The

authors show that recommendation quality improves when the derived interest patterns are taken into account. The latter formalises collaborative filtering as a time-dependent, iterative prediction problem over a dynamic, growing dataset. The authors propose adaptive temporal collaborative filtering, a method of temporally adapting the size of user kNN neighbourhoods based on the performance measured up to the current time.

Another method that adopts simple heuristics to improve collaborative filtering is proposed in [5], where the authors describe a time-biased kNN algorithm exploiting only the most recent ratings from the neighbours. It showed better performance than other kNN recommendation strategies.

Other approaches adapted factorization algorithms to face with temporal effects. Remarkable examples of these methods are described in [14] and [21]. Koren [14] suggested that a mere decay of older instances or usage of separate models for tracking the evolution of preferences cause a loss of prediction accuracy. The proposed solution is to model the temporal dynamics along the whole time period, allowing to separate volatile factors from durable ones, in order to capture the way user and product characteristics change over time. Xiong et al. [21] proposed a factorization method based on probabilistic latent factor models. In addition to the factors that are used to characterize entities, the authors introduce another set of latent features for each different time period. These additional factors represent the preference of latent features at each particular time, so that they are able to capture the evolution of preferences.

Compared to these work which handled the temporal dynamics in different ways, we focus on explicitly modeling short-term preferences and their influence on recommendation of items that will shortly be consumed.

Among approaches that propose different models for long-term and short-term preferences, Cantador et al. [7] designed a content-based news recommender system in which short-term preferences are inferred from the click history, and final ranking of items is adapted to the current context of interest. In [20], the authors propose propose a graph-based approach that introduces session nodes, associated with a user at specific time, to capture short-term linkages between items. If two items are connected by session nodes, their similarity is assumed to be contributed by short-term preferences. An algorithm for preference fusion is designed for temporal recommendation, that proved to be effective on real datasets.

Differently from previous methods, our short-term model tries to capture *semantic* similarity among items by looking at their content, rather than simply co-occurence within sessions or clicking history, with the hope that the semantic approach, combined with time information, helps to discover short-term *relatedness* among them.

3 Time-Aware Distributional Recommender System

Models of Distributional Semantics have drawn a lot of attention in recent years due to their capability of capturing semantics at a latent level, without the re-

quirement of learning algorithms or human-edited resources. Such models build a vector space of meanings where concepts are represented through vectors and the relatedness between meanings is expressed through a proximity function of the points they are represented by. Usually these models are built by skimming a large corpus in order to gather information about distribution of words in a text. Indeed, statistics about word co-occurrences are useful to infer paradigmatic relationships among words, i.e. relations about words that can be used interchangeably. One of the commonest use of such a model is for computing the similarity between words, since the vector components grasp the semantic of word usage in context. Then, the vector addition between words belonging to a text is an easy way to extend to a whole sentence/paragraph/document such a similarity. This work exploits the idea of adapting Distributional Semantic Model (DSM) to item descriptions as a unified framework for both representing the semantic content of items and computing the similarity between them.

3.1 DSM-based Recommender System

The distributional semantic-based recommender system relies on Random Indexing (RI) [13] for building up the semantic space. Given a text corpus, RI technique consists of the following two steps:

1. A *random vector* is assigned to a term in the corpus vocabulary. This vector is highly dimensional, with very few elements -called *seed*- that take values in $\{-1, 1\}$. The dimension of the reduced space corresponds to the random vector dimensionality;
2. The *semantic vector* representation for the term is built by analyzing the whole corpus and summing the random vectors of co-occurring terms in a given text window.

Mathematically, the sum over random vectors corresponds to multiply the original co-occurrence matrix by a projection operator, which preserves the distance proportion between points. The resulting space, called *WordSpace* has two strenght points: 1) the reduced dimension enables a quicker computation of similarities, and 2) likewise Latent Semantic Analysis, shrinking the number of components to a smaller set of contexts makes high order relationships more prominent.

The item space is built upon the previously computed *WordSpace*: the item representation comes from the sum of semantic vectors associated to the item textual content. The user profile, in turn, is built on the basis of the item vector representations she liked or disliked before. In particular, the model keeps trace of both positively and negatively rated items and builds two different profile vectors, u^+ and u^-, as the sum of positive and negative items, respectively. In order to get a single profile vector on which basis the model will compute the recommendations, we exploit the orthogonal projection operator, which has been successfully employed in both retrieval and recommendation scenarios [2, 18]. The idea behind the use of orthogonalization is that if two vectors are

orthogonal with respect to each other, they do not share components, which translates into "they have unrelated concepts". Hence, if we want to express the user profile through a vector that reflects the positively rated items while discarding for negative ones, logically we should represent the user vector u as: $u_1^+ \vee u_2^+ \vee \ldots \vee u_n^+ \wedge NOT(u_1^-) \wedge NOT(u_2^-) \wedge \ldots \wedge NOT(u_m^-)$. However, the logical $aNOTb$ translates into a vector space endowed with a scalar product as the projection of a onto the orthogonal space $\langle b \rangle^\perp \equiv \{v \in V : \forall b \in \langle b \rangle, v \cdot b = 0\}$, where $\langle b \rangle$ is the subspace $\{\lambda b : \lambda \in \mathbb{R}\}$. Thus, computing u corresponds to summing all items in u^+ and then projecting this vector onto the orthogonal space generated by the vectors in u^-. However, following the De Morgan rules, the computation of u can be semplified in $u = u_1^+ \vee u_2^+ \vee \ldots \vee u_n^+ \wedge NOT(u_1^- \vee u_2^- \vee \ldots \vee u_m^-)$, which corresponds to the orthogonalization of two vector (the sum of positive and negative items) performed through the Gram-Schmidt method. Then, the model of recommendation consists in exploring the set of non rated items, in order to assess their similarities with respect to the computed user profile vector. Such a similarity is computed as the cosine similarity, then the ranked list of recommended items can be presented to the user.

3.2 Time-adaptive algorithm

A time-adaptive user profile should be able to grasp changes in the user's behaviour in order to reflect the latest user tastes. We tackle this problem by profiling the user preferences with respect to a sliding window of time: i.e. the items which contribute to building the profile are those occurred shortly before the recommendation. This kind of short-term model was initially proposed by Billsus and Pazzani [3]. However, the user may occasionally manifest a burst of interest towards new items, then by giving more prominency only to the latest voted items can result in suggestions that diverge from the real user's preferences. The time-adaptive algorithm we propose aims to reflect the recency of user's interests in the recommendation process without completely neglecting the role of items that belong to the remote history of user. Indeed, the set of items (*profile set*) which contribute to building the user's profile is collected by taking into account two factors:

Time: Recent items contribute more to the profile;
Similarity: The profile set tends to preserve the items whose content is similar to the newly added one.

Let $I = i_1, ..., i_k$ be such a profile set, every time the user rates a new item i_{new}, this is automatically added to I, while an older item is removed. The element to be discarded is selected as follows:

$$i_{old} = argmin\{i \in I, sim(i, i_{new}) \cdot e^{-\lambda \cdot (t_{new} - t_i)}\} \quad (1)$$

where t identifies the rating time, for both the new item and that under observation, sim is the similarity between two items computed in the item space described in the previous subsection, and λ is a decay factor. Equation 1 aims

to eliminate the item most dissimilar from the newly introduced one. However, in doing so we try to keep coherent the user profile by weighting this factor with the exponential function, whose role is to smooth similarity through time. Then, two possibilities may occur:

1. The new item completely diverges from the user history. In this case all items in I will take on a small similarity and the exponential function will contribute more to equation 1;
2. The new item has some degree of similarity with some items in I. In this case the similarities will be reduced by the exponential function which serves to mediate the contribute of those similar items rated a long time ago.

Among these, the first scenario is the most interesting, since it reflects a potential new trend in the user's preference. Under this condition, the effect of equation 1 on a new incoming item would be that of consolidating this new trend in the user profile, if a newly added item is similar to that latest one, or quickly discard the "exception", thank to the contribute of the exponential function. The λ parameter plays in this context an important role, since it regulates how fast the exponential function has to reduce its rate [10].

4 Evaluation

The goal of the evaluation is to assess the capability of the proposed time-adaptive algorithm to reflect the recency of user interests without loosing information about consolidated long-lasting preferences. Then, we compare our proposed algorithm (**sim**) with respect to a simple sliding window strategy (**fifo**). In **fifo**, the window of items is kept constant: as a new item is added, the older one is removed, thus following a first-in first-out strategy. This approach discards items on the basis of a mere time factor, and no information about user preferences in long-time is preserved.

4.1 Dataset and system setup

The evaluation is performed on the same dataset proposed by Adomavicius et al. [1] This dataset was originally designed for context-aware recommendation. However, since it contains rating timestamps, it suits our case. This dataset comprises 1,757 ratings from 117 users about 226 movies. However, after the removal of ratings without timestamps, we obtained a total of 1,492 ratings from 51 users on about 218 movies. The *WordSpace* is built by collecting co-occurrences information from a dummy corpus consisting in:

– BNC, a collection of documents from the British National Corpus (BNC)[1], containing 100 million words;

[1] http://www.natcorp.ox.ac.uk/

– CMU Movie Summary Corpus[2], a dataset of 42,306 movie plot summaries extracted from Wikipedia;
– PLOT, a collection of plots of movies in the Adomavicious corpus, extracted from Wikipedia.

The objective behind the use of such a corpus is to provide as wider as possible coverage of all word usages in a language. The *WordSpace* represents the top $150,000$ most frequent keywors, the vector dimension is set up to 400, the number of seeds is of 10 elements, while the window size for computing co-occurrences is of 5 words. The recommender system is implemented in Java, and relies on Lucene API[3] for building both the *WordSpace* and the recommendation model. The factor λ in equation 1 is set to 0.01.

4.2 Evaluation protocol

We evaluate the proposed model on a top-N recommendation task. The evaluation is performed as a time-dependent cross-validation, based on increasing time window [6]. This means that the dataset is split on the basis of temporal order of rated items. For each user, we order ratings on the temporal line, then we choose the first k_1 elements as training and the following k_2 items as test set, where N has to be chosen $\leq k_2$. Then, this methods computes iteratively the training and test set by adding the previous test to the current training set, while the new training set is made by sliding k_2 items along the temporal axis. Each iteration corresponds to a single fold. Then, the evaluation metrics are computed over all the users. We compare the **sim** approach with respect to the **fifo** baseline in terms of Mean Average Precision (MAP) [9] and NDCG (Normalized Discounted Cumulative Gain) [19], two metrics that are particularly suitable for our evaluation since they take into account the order of items in the final rank. In fact, the goal of the evaluation is to assess the ability of the proposed method to suggests items which will shortly be consumed by the user. Therefore, the ranking computed by our recommendation method is compared to the *ideal* ranking defined in the test set by ratings and corresponding time information. In other words, we want to evaluate whether our method is able to rank in the top positions of the recommendation list those items in the test set having high ratings and next to the training set, along the temporal axis.

4.3 Analysis of the Results

We compared the two time-adaptive methods on a variable k_1, i.e. the training set dimension. We set $k_2 = 5$ and $N = 3$, while varing $k_1 \in \{5, \ldots, 15\}$. We decided to use this strategy in order to assess the validity of the proposed method when a wider profile set is available.

Tables 1a and 1b report the results of the evaluation with respect to MAP and NDCG metrics. Both tables show a similar trend. The best result is obtained on

[2] http://www.ark.cs.cmu.edu/personas/
[3] http://lucene.apache.org/

$k_1 = 7$ by **sim** method, which gives also better overall figures. There are only two exceptions to this trend, with $k_1 = 11$ and $k_1 = 14$, although the differences in reported values are very small. Another common trend showed by results is that, as we increase the k_1 dimension, the differences between **sim** and **fifo** become smaller. However, we ascribe such a trend to the fact that by increasing the dimension of the training set, the differences between **fifo** and **sim** strategies become smaller since the variability in the set decreases (i.e. an increasingly number of items from the past of the user become part of the training set).

Table 1: Evaluation results at different size of k_1.

(a) MAP

k_1	**fifo**	**sim**
5	0.447	0.460
6	0.444	0.469
7	0.465	0.499
8	0.456	0.461
9	0.459	0.469
10	0.457	0.466
11	0.458	0.455
12	0.459	0.468
13	0.449	0.457
14	0.460	0.458
15	0.431	0.451

(b) NDCG.

k_1	**fifo**	**sim**
5	0.555	0.569
6	0.552	0.575
7	0.569	0.597
8	0.559	0.564
9	0.560	0.575
10	0.566	0.573
11	0.562	0.557
12	0.558	0.565
13	0.550	0.557
14	0.567	0.564
15	0.537	0.555

5 Conclusion and Future Work

Generally, the behavior of a user may be determined by her long-term interests, but at any given time, she is also affected by short-term preferences or information needs. In this paper, we argue that time is not the only factor to be taken into account in order to distinguish among short-time and long-time interests. We started from the classical sliding window approach and suggested that older ratings do not correspond definitely to older preferences, but content of the items should be considered as well. We proposed an approach that models short-term preferences by adopting a content-based sliding window approach: when a new ratings comes into the system, the replacement of an older one is performed by taking into account both a decay function for user interests and content similarity between items on which ratings are provided, computed by distributional semantics models. We compared the proposed approach to the simple FIFO strategy (the new rating replaces the oldest one). Experimental results confirmed the hypothesis that the gradual decay of influence of ratings,

combined with content similarity between items, is actually useful for modeling short-term preferences, especially when a few ratings are available to train the system. As a future work, we plan to evaluate our short-term model on a wider dataset. Furthermore, we want to design a model for long-term preferences, as well as a way to integrate the two models in order to have an overall recommendation strategy.

Acknowledgements

This work fulfils the research objectives of the PON 02_00323_2938699 CUP B86D130000300007 project "EFFEDIL - SOLUZIONI INNOVATIVE PER L'EFFICIENZA ENERGETICA IN EDILIZIA" funded by the Italian Ministry of University and Research (MIUR).

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (Jan 2005)
2. Basile, P., Caputo, A., Semeraro, G.: Negation for document re-ranking in ad-hoc retrieval. In: Amati, G., Crestani, F. (eds.) Advances in Information Retrieval Theory, Lecture Notes in Computer Science, vol. 6931, pp. 285–296. Springer Berlin Heidelberg (2011)
3. Billsus, D., Pazzani, M.J.: User modeling for adaptive news access. User Model. User-Adapt. Interact. 10(2-3), 147–180 (2000), http://dx.doi.org/10.1023/A:1026501525781
4. Billsus, D., Pazzani, M.J.: The adaptive web. chap. Adaptive News Access, pp. 550–570. Springer-Verlag, Berlin, Heidelberg (2007), http://dl.acm.org/citation.cfm?id=1768197.1768218
5. Campos, P.G., Bellogín, A., Díez, F., Chavarriaga, J.E.: Simple time-biased knn-based recommendations. In: Proceedings of the Workshop on Context-Aware Movie Recommendation. pp. 20–23. CAMRa '10, ACM, New York, NY, USA (2010)
6. Campos, P., Dez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Modeling and User-Adapted Interaction 24(1-2), 67–119 (2014), http://dx.doi.org/10.1007/s11257-012-9136-x
7. Cantador, I., Bellogín, A., Castells, P.: Ontology-based personalised and context-aware recommendations of news items. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01. pp. 562–565. WI-IAT '08, IEEE Computer Society, Washington, DC, USA (2008), http://dx.doi.org/10.1109/WIIAT.2008.204
8. Cao, H., Chen, E., Yang, J., Xiong, H.: Enhancing recommender systems under volatile userinterest drifts. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. pp. 1257–1266. CIKM '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1645953.1646112
9. Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Addison-Wesley Publishing Company, USA, 1st edn. (2009)

10. Ding, Y., Li, X.: Time weight collaborative filtering. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. pp. 485–492. CIKM '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1099554.1099689
11. Ding, Y., Li, X., Orlowska, M.E.: Recency-based collaborative filtering. In: Proceedings of the 17th Australasian Database Conference - Volume 49. pp. 99–107. ADC '06, Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2006), http://dl.acm.org/citation.cfm?id=1151736.1151747
12. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA. pp. 230–237. ACM (1999), http://doi.acm.org/10.1145/312624.312682
13. Kanerva, P.: Sparse Distributed Memory. MIT Press (1988)
14. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 447–456. KDD '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1557019.1557072
15. Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM 53(4), 89–97 (Apr 2010), http://doi.acm.org/10.1145/1721654.1721677
16. Lathia, N., Hailes, S., Capra, L.: Temporal collaborative filtering with adaptive neighbourhoods. In: Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 796–797. SIGIR '09, ACM, New York, NY, USA (2009)
17. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 73–105. Springer (2011), http://dx.doi.org/10.1007/978-0-387-85820-3_3
18. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Random indexing and negative user preferences for enhancing content-based recommender systems. In: Huemer, C., Setzer, T. (eds.) E-Commerce and Web Technologies, Lecture Notes in Business Information Processing, vol. 85, pp. 270–281. Springer Berlin Heidelberg (2011)
19. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer US (2011)
20. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long- and short-term preference fusion. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 723–732. KDD '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1835804.1835896
21. Xiong, L., Chen, X., Huang, T., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA. pp. 211–222. SIAM (2010), http://dx.doi.org/10.1137/1.9781611972801.19
22. Zimdars, A., Chickering, D.M., Meek, C.: Using temporal data for making recommendations. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. pp. 580–588. UAI '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001), http://dl.acm.org/citation.cfm?id=647235.720264