Modeling the Dynamics of User Preferences in Coupled Tensor Factorization

Dimitrios Rafailidis*
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
draf@csd.auth.gr

Alexandros Nanopoulos
School of Management
Catholic University of Eichstät-Ingolstadt
Ingolstadt, Germany
alexandros.nanopoulos@ku.de

ABSTRACT

In several applications, user preferences can be fairly dynamic, since users tend to exploit a wide range of items and modify their tastes accordingly over time. In this paper, we model continuous user-item interactions over time using a tensor that has time as a dimension (mode). To account for the fact that user preferences are dynamic and change individually, we propose a new measure of user-preference dynamics (UPD) that captures the rate with which the current preferences of each user have been shifted. We generate recommendations based on factorizing the tensor, by weighting the importance of past user preferences according to their UPD values. We additionally exploit users' side data, such as demographics, which can help improving the accuracy of recommendations based on a coupled, tensor-matrix factorization scheme. Our empirical evaluation uses a real data set from last.fm, which allows us to demonstrate that user preferences can become very dynamic. Our experimental results show that the proposed method, by taking into account these dynamics, outperforms several baselines.

Categories and Subject Descriptors

H.2.8 [**Information Systems Applications**]: Miscellaneous

Keywords

Preference dynamics; Coupled tensor factorization.

1. INTRODUCTION

Users may change their interest over time, especially in applications where they interact customarily with a wide range of items when consuming news or music online. For this reason, recent research started to incorporate temporal effects into model-based collaborative filtering [2, 6, 8], by observing drifts in rating behavior and modeling the way user and item characteristics change over time (e.g., users liking cartoons when they are young, but disliking them when growing up). However, such approaches do not take into account the facts that in several applications: (i) users interact continuously with items over time [3], e.g., when consuming music online or purchasing several categories of retail products (e.g., household items); or (ii) that implicit quantitative feedback may be provided by users, such as the number of times they listened to a song or downloaded a video within a time period (in contrast, [3] focused on explicit feedback in the form of ratings).

Continuous user-item interactions over time can be captured with a sparse tensor whose dimensions (modes) correspond to user, items, and time. Each non-empty tensor cell can record explicit or implicit (quantitative) feedback within the given period. For instance, Xio et al. [7] used a Bayesian probabilistic tensor factorization model for movie recommendation. Spiegel et al. [5] used tensor factorization on evolving data with different time-periods being modeled as time slices in the tensor. They proposed the use of a Smoothing Factor (SF), by exponentially decreasing the importance of past preferences. Nevertheless, these approaches ignore the fact that changes in user preferences can vary individually, i.e., some users' tastes may be very volatile whereas others may keep their tastes relatively stable over time.

In this paper, we follow the aforementioned approaches and model varying user preferences with tensors. We consider individual changes in preferences based on a new measure of user-preference dynamics (UPD) that captures the rate with which the current preferences of each user have shifted compared to his past preferences. Recommendations are generated based on tensor factorization, by weighting the importance of past user preferences according to their UPD values. Motivated by recent research on auxiliary information (a.k.a. side information) [4], we examine users' side data, such as demographics, to improve the recommendation accuracy based on a coupled tensor factorization model, which captures shared latent factors across the time-varying user preference data and the side information about the users. Our empirical evaluation uses a real data set from last.fm, which demonstrates that user preferences can become very dynamic. Our experimental results show that the proposed method outperforms several baselines.

2. THE PROPOSED MODEL

The inputs of the proposed model are (1) a third-order tensor $\mathcal{X} \in \mathbb{R}^{|U|\times|I|\times|T|}$, where U, I and T denote the sets

*Also with the Comp. Science Dept. of Cyprus Univ.

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645758 .

of users, items and time-periods, respectively and each non-empty tensor cell $\mathcal{X}(u,i,t)$ contains the number of interactions of user $u \in U$ to item $i \in I$ at the time-period $t \in T$; (2) an auxiliary matrix $Y \in \mathbb{R}^{|U|\times|D|}$ of users' side information with a set of D attributes, coupled in the user dimension (mode) U of tensor $\mathcal{X}$; and (3) the rank R of $\mathcal{X}$. Firstly, we model the temporal and users' side information (Section 2.1), and then the proposed model reweigh the past interactions of tensor $\mathcal{X}$ based on the users' recent interaction behavior to items(Section 2.2). The output of the proposed model is a low rank R approximation of $\mathcal{X}$, denoted by $\hat{\mathcal{X}}$, using a CANDECOMP/PARAFAC (CP) decomposition (Section 2.3).

2.1 Modeling Changing User Preferences and Side Information

Each non-empty tensor cell $\mathcal{X}(u,i,t)$ contains the number of interactions of user u to item i at the time period t (assuming implicit quantitative feedback). The time period can be days, months, semiannuals (i.e., 6 months), or whole years, corresponding to the $|T|$ different time slices $|U| \times |I|_t$ of tensor $\mathcal{X}$, with $t \in 1 \dots |T|$. The choice of the time period mainly depends on the application of the recommender system. This means that the test time period, e.g., the test month where the personalized recommendations have to be generated, is included in the last time slice $|U| \times |I|_{|T|}$, e.g., the current semiannual, of the tensor $\mathcal{X}$.

With respect to the users' side information the u-th, $u \in U$, row of the auxiliary matrix Y corresponds to the set D of attributes of user u. In case of numerical attributes, e.g., age, we perform an equal-width binning, where for each numerical entry in the matrix Y we store the respective number of the bin. In the case of categorical attributes, such county and gender, we calculate the c distinct categorical values and then we create a binary vector $\mathbf{c}_i \in \mathbb{R}^{c\times 1}$, where 1 denotes the categorical attribute of user u_i. Finally, we append the transformed numerical and categorical attributes to generate the final $|D|$ attributes in matrix Y.

2.2 Users Preference Dynamics (UPD)

Given a test period $t+1$ within the current/last time slice $|U| \times |I|_{|T|}$ of the tensor $\mathcal{X}$, e.g., the test month in the last semiannual, for each user u we calculate the UPD$_u$ value as follows:

$$\mathrm{UPD}_u = 1 - \frac{|I^u_{cur} \cap I^u_{prev}|}{|I^u_{cur} \cup I^u_{prev}|} \tag{1}$$

where $I^u_{cur} \subseteq I$ denotes the set of items that user u has interacted at the current/last time slice $|U| \times |I|_{|T|}$ of $\mathcal{X}$ and $I^u_{prev} \subseteq I$ is the union set of the items that user u has interacted at all the previous time slices $|U| \times |I|_t$, with $t = 1 \dots |T| - 1$. The numerator of the fraction in Eq. 1 is the number of common items that user u has interacted at the current and the previous time periods, whereas the denominator of the fraction is the number of the distinct items that user u has interacted overall. According to Eq. 1, low UPD$_u$ values indicate that user u preserved his preferences, whereas high ones correspond to users' u high tendency to change his preferences at the current time period (the $|T|$-th time slice of $\mathcal{X}$). After calculating the $|U|$ different UPD values, we decrease the weights of each interaction of user u at the $|T| - 1$ different past time slices $|U| \times |I|_t$, by multiplying them with the smoothing factor $sf_u = 1 - \mathrm{UPD}_u$, as follows $\forall u \in U$ and $|T| > 1$[1]:

$$\mathcal{X}(u,i,t) := sf_u * \mathcal{X}(u,i,t) \tag{2}$$

$$\text{with } t = 1, \dots, |T| - 1 \text{ and } i \in I^u_{prev}$$

We have to mention that the smoothing factor sf_u is different for each user u, depending on the users' current interaction behavior to items within the current time period. This comes in contrast to several works such as [5], where the smoothing factor is applied in an non-personalized way. The outcome of this process is the recalculation of the respective entries (number of interactions) of $\mathcal{X}$ based on Eq. 2.

2.3 The Low Rank R Approximation Of The Tensor $\mathcal{X}$

In our model we used the CP decomposition of $\mathcal{X}$ with the recalculated entries based on Eq. 2. To generate the output of the proposed model, we calculate the $\hat{\mathcal{X}}$ low rank R approximation as follows:

$$\hat{\mathcal{X}} = \sum_{r=R}^{R} \mathbf{u}_r \circ \mathbf{i}_r \circ \mathbf{t}_r \tag{3}$$

where $\circ$ is the outer product operation and $\mathbf{u}_r$, $\mathbf{i}_r$, $\mathbf{t}_r$ are the respective user, item, time vectors of the r-th component. The goal of the proposed model is to minimize the low rank approximation error $||\mathcal{X} - \hat{\mathcal{X}}||^2_F$, where $||\cdot||_F$ indicates the Frobenius norm. The R component of a tensor $\mathcal{X}$ and the auxiliary matrix Y is defined as:

$$f(A,B,C,V) = ||\mathcal{X} - [A,B,C]||^2 + ||Y - AV^T||^2 \tag{4}$$

where matrices $A \in \mathbb{R}^{|U|\times R}$, $B \in \mathbb{R}^{|I|\times R}$ and $C \in \mathbb{R}^{|T|\times R}$ are the factor matrices of $\mathcal{X}$, extracted by CP; matrices A and $V \in \mathbb{R}^{|D|\times R}$ are the factor matrices extracted from matrix Y (with users' side information) by a matrix factorization technique. Also, in Eq. 4 we use the notation $\hat{\mathcal{X}} = [A,B,C]$ to denote the low rank approximation based on the CP model. Our goal is to calculate the matrices A, B, C and V that minimize the objective function of Eq. 4. In order to solve the minimization problem we used the CMTF-OPT algorithm of the CMTF toolbox[2], which also handles missing data, since $\mathcal{X}$ is usually sparse.

The final top-k recommendations for each user u at the current test time-period t are generated by ordering in descending order the entries of $\hat{\mathcal{X}}(u,:,t)$, i.e. the respective column of items with indices $u \in U$ and t within the current/last time slice of $\hat{\mathcal{X}}$. Thus, the outcome of the proposed model is the respective top-k (recommended) items.

3. EXPERIMENTS

3.1 The last.fm Dataset

In our experiments, we used the last.fm Dataset - 1K users[3] which contains the listening habits for $|U|$=992 users. The dataset consists of tuples in the form of user-artist-song-timestamp over 54 months (till May, 5th 2009). In to-

[1] For $|T|$=1, $\forall u \in U$ $sf_u = 1$

[2] www.models.life.ku.dk/joda/CMTF_Toolbox/

[3] www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/

tal, there are $|I|$=176,948 artists[4] and 19,150,868 listening events, corresponding to track-listenings. The distribution of the listening events is presented in Figure 1. In our experiments, we split the dataset into 9 time periods (time slots $S_1 \ldots S_9$), corresponding to 9 semiannuals. Thus, we have $|T|$=9 different time slices in the tensor, where each slice corresponds to a six-months period. In this dataset, users' side information is also available, including age, gender, country (with missing values, however). Users in this dataset come from 68 different countries. Since gender and country are categorical values, we used the transformation technique of Section 2.1 to generate $|D| = 71$ attributes in total.

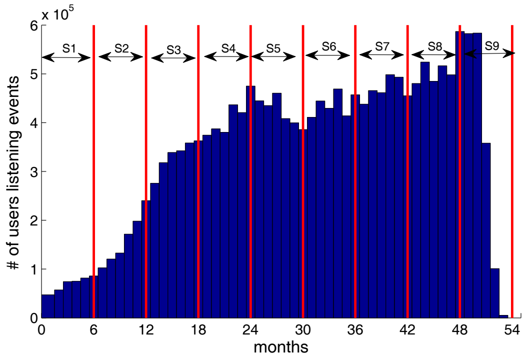

Figure 1: Users listening events (track-listenings) in the last.fm dataset.

3.2 Evaluation Protocol

Tuples were transformed in the form of user-artist-time slot-# of listening events, corresponding to how many times (number of interactions) a user has listened tracks of an artist within the time slot. In doing so, we generated 2,963,653 tuples. Given a set of training months the goal of the proposed model is to perform top-k artist recommendation for a user at a test month. In our experiments we used a time window equal to semiannual, where as training set we considered all the past months of the previous semiannuals and the first five months of the current semiannual. The goal is to predict the artists that each user is going to listen at the last (6-th) test month of the current semiannual. Therefore, we have nine different test sets of tuples at test months 6, 12, 18, 24, 30, 36, 42, 48 and 54, denoted by red lines in Figure 1, and nine different training sets of tuples at the respective past months. Since we noticed in the dataset that each user does not listen to more than 100 artists for a test month, in our experiments we report Average Recall (AR), with k=100 artists.

3.3 Preference Dynamics

In Figures 2 and 3, we group users from the test months into 3 different groups based on their listening events and their UPD value (Eq. 1), respectively. From Figure 2 we can observe that users increase the number of their listening events over time. The evolution of the 3 different groups based on the UPD metric in Figure 3 shows that users tend to significantly shift their preferences over time, since the percentage of users in the group UPD $\geq$ 0.75 is highly increased over time. According to Eq. 1, the user group with UPD $\geq$ 0.75 contains the users at the test months that have listened more than a 75% percentage of new artists than they have listened at the past months. An interesting observation is that we have a critical point at 18 months, where users start to significantly shift their preferences, denoted by the starting point of the high increase of the percentage of users in group UPD $\geq$ 0.75 and the starting point of the high decrease of groups UPD $\leq$ 0.5 and 0.5 $<$ UPD $<$ 0.75 .

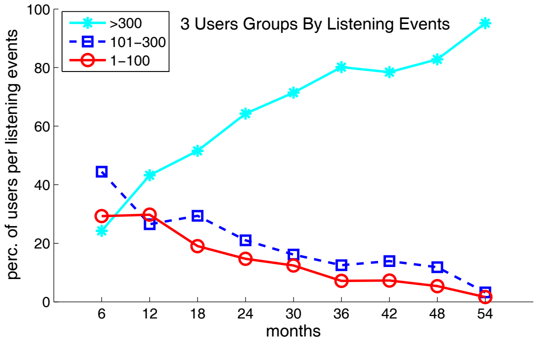

Figure 2: The evolution of the 3 different users groups based on the listening events.

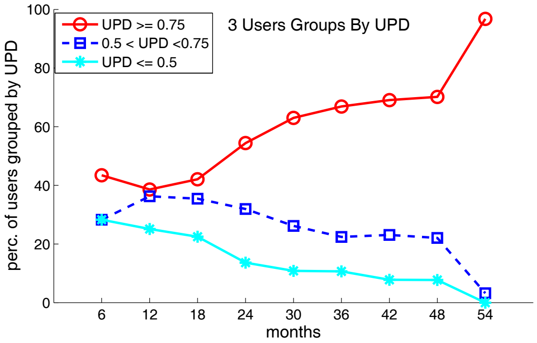

Figure 3: The evolution of the 3 different users groups based on UPD.

3.4 Settings

In our experiments, we evaluated the proposed model, by considering (a) only the modeling of side information with the CP decomposition, performing thus Coupled Tensor Factorization (**CTF**) and (b) the combination of side information with the weighting scheme based on UPD in Eq. 1 and 2 (**UPD-CTF**). As competitive methods we consider the **TF** method of [1] and the **SF** method of [5]. For **SF**, we varied the smoothing factor from 0.1 to 0.9 and selected the value 0.3 as best performing. Lower values of the smoothing factor mean that the SF method down-weigh more the past preferences, in our case listening events. As **baseline** method, we used the most-popular artists methods, which recommends the top-k most popular artists for each user within the training months.

Since determining the rank R of the tensor is a NP-problem, we varied R by 5, 10, 15 and 20. We concluded to R=15, since a further increase of rank R results in a higher computational cost without paying off in terms of recommendation accuracy. TF and SF method have similar computational complexities, whereas CTF and UPD-CTF have higher com-

[4] To remove extreme sparsity for artists, we applied the p-core filtering technique, with p =0.2%$|I|$.

putational complexity for the required coupling of matrix Y with the users' side information. For the methods (a) with and (b) without the coupling the computational times (for R=15) are 126 and 432 secs, respectively (on a Windows 7 PC with Intel core i7 2700K at 3.50 GHz, 8GB Ram).

3.5 Results

The results on the nine test months are presented in Figure 4. The proposed UPD-CTF method outperforms TF, SF and the baseline, because it captures users' preference dynamics in a personalized way and also incorporates auxiliary information. Using paired t-test we found that the difference between the reported results for the proposed UPD-CTF method against the competitive approaches is statistically significant at level 0.05. An important observation is that after the critical point of 18 months, where users start to significantly shift their preferences (see description in Section 3.3), the accuracy of TF and SF starts to decrease fast, even lower than the baseline method. This happens because both methods neither handle the users' personalized preference dynamics nor use auxiliary information. Finally, an interesting observation is that, despite the fact that CTF does not handle users' preference dynamics, it preserves the recommendation accuracy relatively high, which means that users' auxiliary information plays a crucial role in recommender systems.

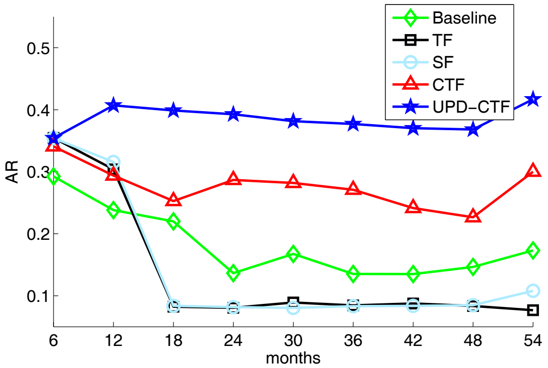

Figure 4: Methods comparison.

In Table 1 we report the performance of the examined methods separately for the three users groups based on UPD for the test months 12 and 54. We selected these months because they are before and after the critical point of 18 months, where the users preference dynamics (UPD) start to change significantly. As expected the baseline method performs worse in the case of users in the groups of UPD $\geq$ 0.75 and 0.5 $<$ UPD $<$ 0.75 than users in group UPD $\leq$ 0.5, since in the latter group we have users that remain stable to their preferences over time. Additionally, the proposed UPD-CTF method preserves the recommendation accuracy high in all 3 different groups of users based on UPD, by handling the users' preferences dynamics and exploiting side information.

Table 1: Methods performance for the 3 different users groups based on UPD (test months 12 and 54).

12 months	UPD $\geq$ 0.75	0.5 $<$ UPD $<$ 0.75	UPD $\leq$ 0.5
Perc. users	38.6%	36.28%	25.12%
Baseline	0.21 ± 0.02	0.24 ± 0.09	0.26 ± 0.06
TF	0.30 ± 0.13	0.32 ± 0.08	0.27 ± 0.11
SF	0.31 ± 0.13	0.34 ± 0.09	0.28 ± 0.13
CTF	0.27 ± 0.12	0.31 ± 0.08	0.26 ± 0.12
UPD-CTF	$\mathbf{0.41 \pm 0.11}$	$\mathbf{0.42 \pm 0.06}$	$\mathbf{0.4 \pm 0.08}$
54 months			
Perc. users	96.77%	3.23%	0%
Baseline	0.16 ± 0.09	0.30 ± 0.12	N/A
TF	0.09 ± 0.08	0.06 ± 0.05	N/A
SF	0.11 ± 0.06	0.07 ± 0.07	N/A
CTF	0.32 ± 0.12	0.34 ± 0.08	N/A
UPD-CTF	$\mathbf{0.43 \pm 0.09}$	$\mathbf{0.42 \pm 0.11}$	N/A

4. CONCLUSIONS

We modeled users preference dynamics and users' side information in coupled tensor factorization. Our experiments showed that the proposed model can achieve significant improvement in recommendation accuracy against competitive methods. This happens because our model considers continuous user-item interactions over time, by capturing the rate with which user preferences change, and also exploits user's side information. Since for each test time-period we have to recalculate the approximation of the tensor offline, we plan to extend the proposed model to incremental tensor factorization techniques. Another research direction is to address the cold-start problem when computing change rates for users without adequate number of interactions.

5. REFERENCES

[1] D. M. Dunlavy, T. G. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *TKDD*, 5(2):10, 2011.

[2] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of KDD*, pages 447–456, 2009.

[3] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 210–217, 2010.

[4] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima. Tensor factorization using auxiliary information. In *Proceedings of ECML/PKDD*, 2011.

[5] S. Spiegel, J. H. Clausen, S. Albayrak, and J. Kunegis. Link prediction on evolving data using tensor factorization. In *PAKDD Workshops*, pages 100–110, 2011.

[6] L. Xiang and Q. Yang. Time-dependent models in collaborative filtering based recommender system. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 450–457, 2009.

[7] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the SIAM International Conference on Data Mining*, pages 211–222, 2010.

[8] C. Zhang, K. Wang, H. Yu, J. Sun, and E.-P. Lim. Latent factor transition for dynamic collaborative filtering. In *Proceedings of the SIAM International Conference on Data Mining*, SDM '14, pages 452–460, 2014.