Repeat Consumption Recommendation Based On Users Preference Dynamics And Side Information {Online Technical Report}

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

1. EVALUATION DATASETS

1.1 Last.fm

In our experiments, we used the last.fm dataset - 1K users[1] 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 total, there are $|I|$=176,948 artists and 19,150,868 listening events, corresponding to track-listenings. The distribution of the listening events are presented in Figure 1. In our experiments, we split the dataset into 9 time periods (time slots $t = S_1 \dots S_9$), corresponding to 9 semiannuals. Thus, we have $|T|$=9 different time slices in the tensor, where each slice correspond to a six-months period. In the last.fm dataset, the private attributes of users are also available, including age, gender, country, where in many cases the attributes are missing. Users in this dataset come from 66 different countries. Since gender and country are categorical values, we used the transformation technique of Section 2 (for further details please refer to the first step of CTF: *Modeling User Preferences and Side Information*) to generate $|D| = 71$ attributes in total. In the last.fm dataset tuples were transformed in the form of {user, artist, time slot, # of listening events}, corresponding to how many times a user has listened tracks of an artist within the time slot.

Given a set of training months the goal 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 ongoing 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 9 different training sets of tuples at the respective past months. 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. The quality of recommendations is measured as follows, for a test user u that receives a list of k recommended items (top-k list) at a test period t, the recommendation accuracy is defined as the the ratio of the number of relevant artists in top-k list over the total number of relevant artists (all artists in the hidden triplets containing test user u and test period t). Since we noticed in the last.fm dataset that each user does not listen more than 100 artists for a test month, in our experiments we report average recommendation accuracy, with k=100 artists.

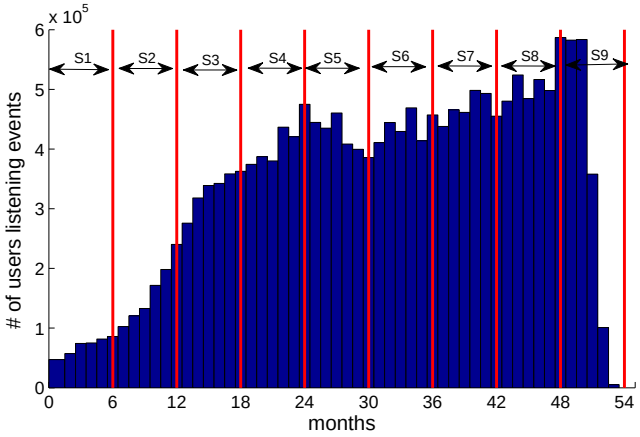


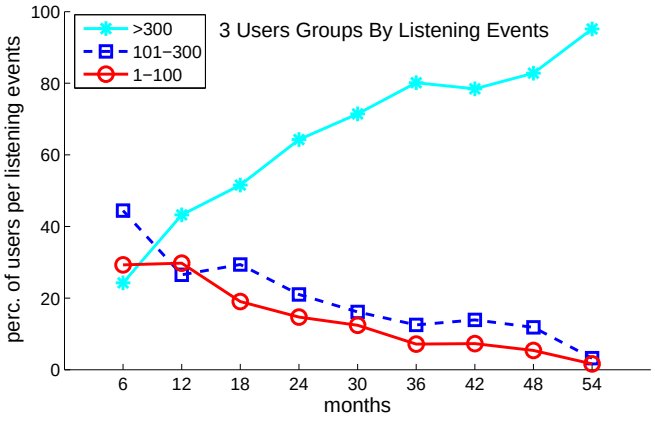
Figure 1: Users listening events (track-listenings) in the last.fm dataset.

For last.fm in Figures 2(a) and 2(b), we group users from the test months into 3 different groups based on their (a) listening events and (b) UPD values. From Figure 2(a) we can observe that users increase their listening events over time, i.e. the percentage of users in group >300 increases over time. Figure 2(b) shows the high variability in how users interact with items in the recommender systems. The evolution of the 3 different groups based on the UPD metric in Figure 2(b) 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. Group $UPD \geq 0.75$ 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

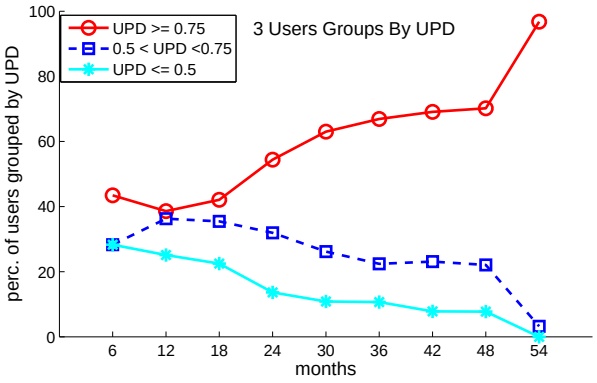
[1] www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/

WWW'15 Companion 18-22 May, Florence, Italy

by the starting point of the high increase of the percentage of users in group $UPD \geq 0.75$ and the high decrease of groups $UPD \leq 0.5$ and $0.5 < UPD < 0.75$.



(a)



(b)

Figure 2: The evolution of the 3 different users groups based on (a) listening events and (b) UPD values in last.fm.

1.2 MovieLens

Also, in our experiments we used the MovieLens - 1M dataset[2] with 1,000,209 anonymous ratings of 3,952 movies of $|U|$=6,040 users who joined MovieLens in 2000. The dataset consists of tuples in the form {user, movie, rating, timestamp} over 36 months. Ratings are made on a 5-star scale, which correspond to users' viewing events, assuming that users have rated movies, after viewing them. The distribution of the viewing events are presented in Figure 3. Accordingly, we split the dataset into 6 time periods (time slots $t = S_1 \ldots S_6$), corresponding to 6 semiannuals, where we have $|T|$=6 different time slices in the tensor, with each slice corresponding to a six-months period. In the MovieLens dataset, the private attributes of users are also available, including age, gender, and 21 occupation types, such as "academic/educator", "lawyer", "doctor/health care", "programmer", etc. Since gender and occupation type are categorical values, we used the transformation technique as in the last.fm dataset, so as to generate $|D| = 24$ attributes in total. Moreover, in the MovieLens dataset we have $|I|$=18 movie-genres, such as "Action", "Adventure", "Animation", "Comedy", "Documentary", "Thriller", "Fantasy", etc. In the MovieLens dataset tuples were transformed in the form of {user, movie-genre, time slot, # of viewing events}, corresponding to how many times a user has watched a movie of a movie-genre within the time slot. The goal is to perform movie-genre recommendation, instead of movie recommendation in the repeat consumption problem, since users rarely watch (interact) with the same movie multiple times. Given a set of training months the goal of the proposed model is to perform top-k movie-genre recommendation for a user at a test month. Similar to the last.fm dataset, 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 ongoing semiannual. The goal is to predict the movie-genre of movies that each user is going to watch at the last (6-th) test month of the current ongoing semiannual. In our experiments we report the recommendation accuracy, with k=3 movie-genres, since in the MovieLens dataset users watch movies from no more than 3 different movie genres for a test month.

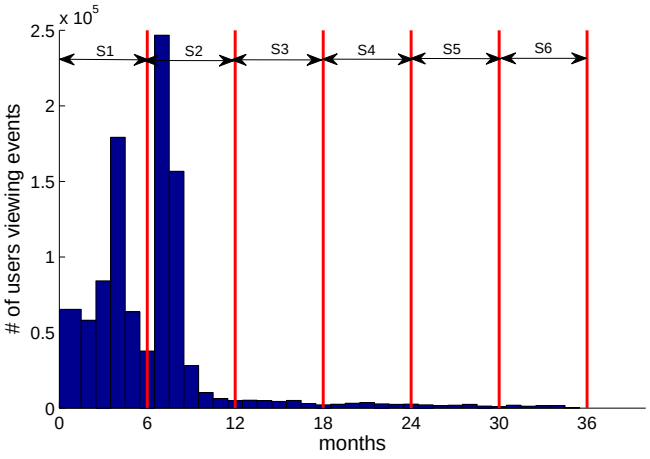
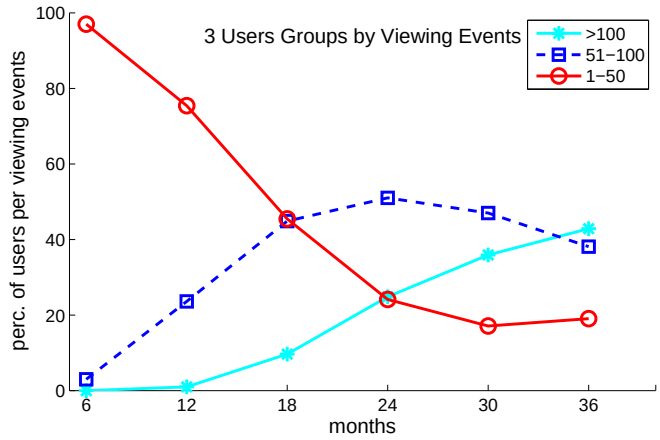


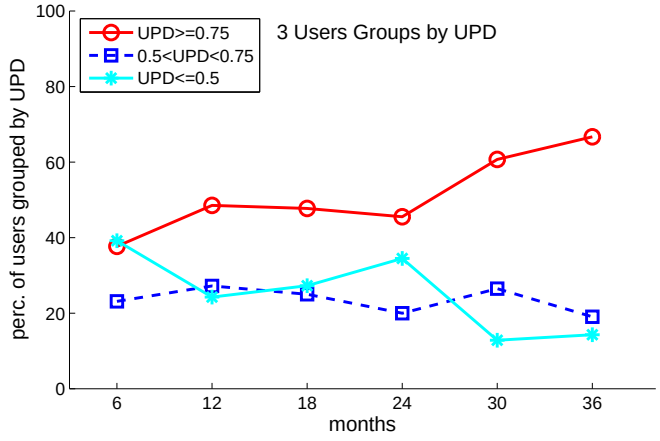
Figure 3: Users viewing events (movies-watched) in the MovieLens dataset.

For MovieLens in Figures 4(a) and 4(b), we group users from the test months into 3 different groups based on their viewing events and their UPD values, respectively. From Figure 4(a) we can observe that users increase their viewing events over time (the increase of users in group >100). Comparing Figure 2(b) of last.fm and Figure 4(b) of MovieLens, users in the MovieLens dataset have less dynamic taste, since the percentage of users in group $UPD \geq 0.75$ is not that highly increased over time as users in group $UPD \geq 0.75$ of last.fm. This means that the percentages of users in MovieLens are more balanced distributed to the UPD groups than users in last.fm. Nevertheless, for test month 24 in MovieLens, we observe a starting point of a slightly increase of the percentage of users in group $UPD \geq 0.75$ and a decrease of the percentages of users in groups $0.5 < UPD < 0.75$ and $UPD \leq 0.5$.

[2] `http://grouplens.org/datasets/movielens/`



(a)



(b)

Figure 4: The evolution of the 3 different users groups based on (a) viewing events and (b) UPD values in MovieLens.

2. PERFORMANCE ON THE 3 UPD GROUPS

In Table 1 we report the performance of the examined methods separately for the three users groups based on UPD for the test months 12/54 and 18/36, for last.fm and MovieLens respectively, selecting months before and after the critical points, where the users preference dynamics 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, both proposed models preserve the recommendation accuracy high in the UPD groups, by handling the users' preferences dynamics and exploiting side information. W-CTF achieves higher recommendation accuracy than CTF for the dynamic UPD groups, since it considers the diversification of the side information by weighting higher the users that have more rare side information.

Table 1: Performance on the 3 UPD groups.

last.fm			
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
CTF	0.41 ± 0.11	0.42 ± 0.06	0.4 ± 0.08
W-CTF	$\mathbf{0.56 \pm 0.09}$	$\mathbf{0.52 \pm 0.07}$	$\mathbf{0.49 \pm 0.06}$
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
CTF	0.43 ± 0.09	0.42 ± 0.11	N/A
W-CTF	$\mathbf{0.55 \pm 0.07}$	$\mathbf{0.53 \pm 0.08}$	N/A
MovieLens			
18 months	$UPD \geq 0.75$	$0.5 < UPD < 0.75$	$UPD \leq 0.5$
Perc. users	47.73%	25%	27.27%
Baseline	0.28 ± 0.01	0.24 ± 0.02	$\mathbf{0.33 \pm 0.03}$
TF	0.34 ± 0.03	0.33 ± 0.02	0.23 ± 0.03
CTF	0.41 ± 0.03	0.38 ± 0.03	0.26 ± 0.03
W-CTF	$\mathbf{0.43 \pm 0.02}$	$\mathbf{0.41 \pm 0.03}$	0.23 ± 0.04
36 months			
Perc. users	66.67%	19.05%	14.29%
Baseline	0.23 ± 0.03	0.28 ± 0.02	$\mathbf{0.32 \pm 0.02}$
TF	0.36 ± 0.04	0.28 ± 0.03	0.14 ± 0.02
CTF	0.48 ± 0.04	0.37 ± 0.03	0.27 ± 0.03
W-CTF	$\mathbf{0.50 \pm 0.03}$	$\mathbf{0.38 \pm 0.03}$	0.24 ± 0.04