# Towards Time-Dependant Recommendation based on Implicit Feedback


Linas Baltrunas
Free University of Bozen-Bolzano,
Via Sernesi 1,
Bolzano, Italy
lbaltrunas@unibz.it

Xavier Amatriain
Telefonica R&D,
Via Augusta 177,
Barcelona, Spain
xar@tid.es



## ABSTRACT

Context-aware recommender systems (CARS) aim at improving user satisfaction to recommendations by tailoring these to each particular context. In this work we propose a contextual pre-filtering technique based on implicit user feedback. We introduce a new context-aware recommendation approach called user *micro-profiling*: The user profile is split into several sub-profiles, each representing the user in a particular context. The predictions are done using these micro-profiles instead of a single user model.

However, the user taste depends on the exact partition of the contextual variable. The identification of a meaningful partition of the user profile and its evaluation is a non-trivial issue, especially when using implicit feedback and a continuous contextual variable.

We propose an off-line evaluation procedure for CARS in these conditions and evaluate our approach on a time-aware music recommendation sytem.


## 1. INTRODUCTION

Recommender systems are powerful tools helping on-line users tame information overload by providing personalized recommendations [10]. Collaborative Filtering (CF) is a successful recommendation technique which automates the so-called "word-of-mouth" social strategy [10]. The music industry is just another example of domains benefiting from recommendation technology. Music consumption is biased towards a few popular artists and recommender systems can help filter, discover and personalize the music that we listen to [4]. The choice of the music during the day is influenced by a contextual conditions, such as the time of the day, mood or a current activity we perform [9], but this type of information is not exploited by standard CF models.

In this work we propose a contextual pre-filtering technique for recommendation called *micro-profiling* (see Section 3 for details). The long-term goal is to make a time-aware recommender system which can accurately predict a user taste, given the current time, i.e., of the day, week or a year. The approach assumes that user preferences change over time but have temporal repetition. For example, a user listens to one type of music while working, and another type of music before going to sleep. The main idea of the approach is to partition a user profile into smaller ones and use these micro-profiles for the prediction instead of a single profile. There are two main challenges for this approach: (1) how to extract meaningful micro-profiles and (2) how to combine them into a single recommendation.

In our experiments, we use implicit information of user taste to infer his preferences. This enables to gather big amounts of time-enriched data without additional user effort. Time is easy to track, does not require additional user input and could be informative enough to determine the user behavior.

However, determining meaningful micro-profiles form implicit feedback over a continuous context variable is a non-trivial problem. The user taste depends on the exact definition of the time slice. For example, imagine a context-aware recommender system which could generate a track recommendation for a user in the morning. The exact definition of morning will influence the final prediction of the algorithm and could be different for each of the user. Moreover, the standard off-line evaluation procedure can not be used for such type of data. Therefore, we propose an off-line evaluation procedure for context-aware recommender systems (CARS) with implicit data and continuous contextual domains (see Section 4).

Context plays important role in determining user behavior providing additional information that can be exploited in building predictive models [2]. Context-aware recommender systems is a new area of research [1]. The approaches can be classified into three main groups: pre-filtering, post-filtering and contextual modeling [2]. User micro-profiling approach falls into the class of pre-filtering algorithms as time is used to alter the original user ratings before making the prediction. The first pre-filtering approach was introduced in the work of Adomavicius et al. [1], where authors extend the classical CF method adding new dimensions representing contextual information. Here recommendations are computed using only the ratings made in the same context as the target one.

In the field of music recommendation, context was reported to improve the prediction accuracy [9, 3]. Jae and Jin [9] used case-based reasoning approach where similarity of cases was extended to include the similarity of the contexts. Authors reported increase of the average precision. Another interesting method, which integrates time into the



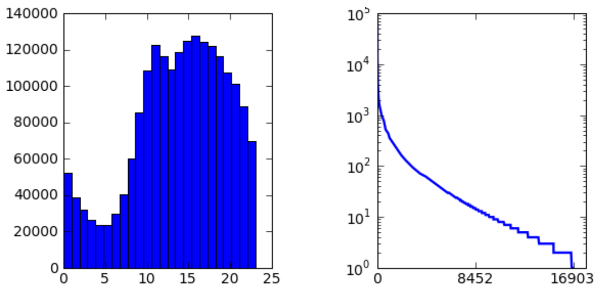
*CARS-2009,* New York, USA

| | |
|---|---|
| #users | 338 |
| #tracks | 322871 |
| #artists | 16904 |
| #entries | 1970029 |
| #ratings (after normalization) | 143091 |
| average mean repetition of a track for a user | 3.09 |
| average mean repetition of an artist for a user | 19.87 |

**Table 1: Summary of the data set**

prediction process of CF recommender system, is presented by Koren [7]. The author created a model based CF tracking the time changing behavior throughout the life span of the data. An idea somewhat related to micro-profiling is explored by Ohbyung and Jihoon [8] where authors present concept lattices to discover context based user profile.

## 2. DATA




(a) Number of tracks listened per hour. (b) Popularity of an artist.

**Figure 1: Last.fm data information**

For our approach we use *implicit* data collected during a two years period (2007-03-01 to 2008-12-31) and containing 338 random Spanish users of last.fm service. Each track is stored into the user's profile together with a time stamp. We cleaned the data from mistyping and removed artists that were listened only by a single user. The summarized information about the data set is listed in Table 1.

The use of implicit feedback data in CF recommender systems raises some challenges on its own [6]. First, the implicit data gives us information only on the positive user feedback i.e., which track or artist she listens most, and when she prefers to listen the artist. However, it misses information on the negative user preferences. This is not the case for other data sets with explicit user ratings, where a user can express positive and negative opinions about an item. Another important issue with implicit user feedback is that the evaluation procedure is not well established (see Section 4 for details).

Furthermore, the music domain requires different techniques from ones used for the movie or book recommendations. Users listen to the the same artist and track many times. Each user in our used data set on average listened for 5828.5 tracks. Repeated consumption of items enables us to analyze a user behavior in different conditions and compare the profile of the same user in various contexts, i.e., weekday versus weekend. Figure 1(a) shows listening behavior of all the users. The users are most active in the afternoon (4 pm.) and least active at 5 am. We also discovered that on average users tend to use last.fm service on working days more than during the weekends. Note that some items are much more popular than others. Figure 1(b) shows how many times an artist was listened in total.

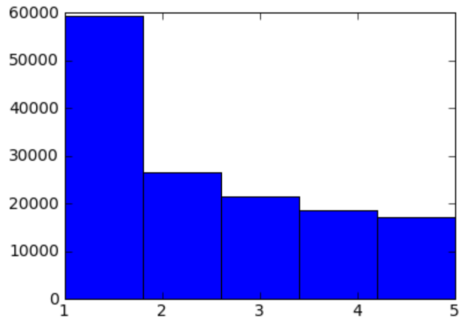



**Figure 2: Rating distribution for the data set.**

Our goal is to build a time-aware RS that can accurately recommend an unkown yet interesting artist (or a song) for the user. In our initial experiments we are recommending an artist rather than a track, therefore, all the mappings are done on the artist level. The ability to recommend a track will be extended in the future.

To measure the performance of the system using the Mean Absolute Error (MAE) we map implicit user feedback into explicit ratings. We use mapping procedure proposed by Celma [4] – which is similar to Hu et al.'s [6]. We use the number of times the user listened to an artist as a proxy of the user preferences. But users' listening habits usually present a power law distribution: That is, a few artists have lots of plays in the user profile, and the rest of artists have much less play counts. Therefore, we compute the complementary cumulative distribution of artist plays in the user profile. Artists located in the top 80-100% of the distribution get a score of 5, artists in the 60-80% range get a 4. Artists with less play counts, in the 0-20% range, will get a rating of 1. In case we have not enough variation in the user profile to divide all counts into 5 groups we assign 3 as the rating.

Figure 2 shows the rating distribution for the data set. Note that we have higher number of artists with small ratings. It is specific property of the music data sets as a single user listens to a large amount of unique tracks or even artists. This leads to many artists that the user has listened only once.

## 3. APPROACH

Our long-term goal is to make a time-aware recommender system which can accurately predict user's music taste, given the current time. The idea is to partition the user $u$ profile into micro profiles $\{u_1, u_2, ..., u_n\}$ that best represents the user in a particular time span. For example, we can have a representation of the same user $u$ in the morning, evening, weekend, summer, etc. Micro-profiles would be more precise model of the user. To make a recommendations we would use multiple micro-profiles instead of a single profile $u$. The rationale behind the approach is that we can improve the accuracy while having a set of coherent and more precise user models. Micro-profiles for a single user can be built for many different time cycles such as day, month or year. One of the challenges, which we shall not approach in this paper, is how to combine the predictions generated for each of the profile and present the final prediction.

But, even a more fundamental problem that we do need to approach, is how to discover meaningful time partition based on the time cycles. Each partition should represent a time slice where user has similar repetitive behavior. For example, working hours of a user if she listens the same set of artists while working. Each user could have different definition of morning and evening and the same time partition might not work globally for all the users.

But, for simplicity and evaluation issues (see Section 4), in this initial work we analyze only non overlapping partitions. Moreover, we evaluate the system by making the predictions without combining several micro-profiles. Finally, we do not look into personalized partitions but rather evaluate global ones. All of these issues should be considered as future work to be addresses.

# 4. EVALUATION CHALLENGES USING IMPLICIT CONTEXT-ENRICHED DATA

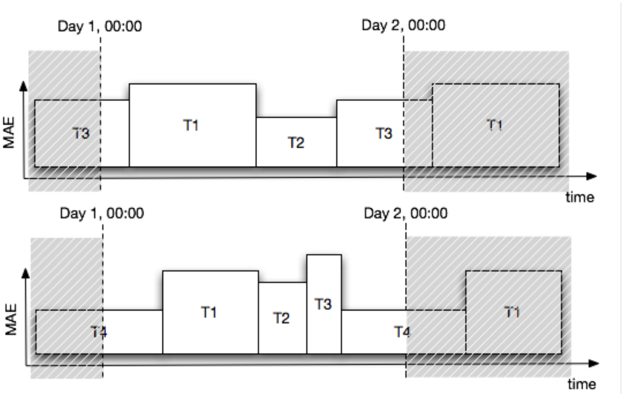



**Figure 3: Examples of the partitioning $T$ for a day.**

The evaluation of a recommender system tries to estimate user satisfaction for a given recommendation. The most common procedure is to use off-line evaluation techniques [5]. To evaluate context-aware recommender systems, different accuracy measures have been used: expected percentile ranking [6], precision, recall, F1 [1], average precision [9]. Most of the previous work on CF evaluates the accuracy of the system using explicit user rankings [6]. In this Section we propose an off-line evaluation technique for implicit user ratings and continuous contextual variables.

In our case, the biggest problem is related to the continuous time variable – in fact, the same evaluation problem generalizes to other continuous contextual variables such as temperature or distance to an object. To the best of our knowledge, this problem has not been faced before since most of the data sets contain ratings with a nominal contextual variable such as companion or weekday [1].

To understand the problem, imagine a scenario where a user continuously listens for music. We want to build a system that would be able to predict her preferences in various times of the day. Lets say the user likes two artists A and B. In the morning she prefers artist A over B. On the contrary, at work she prefers to listen to B more than A. When making a rating prediction for a specific time of the day, we should be able to infer these relations. Interestingly, the exact partitioning of the time domain defines the ground truth which we want to predict. For example, if we define our "morning" to be the time interval from 6 am to 9 am we will infer the user preferences by counting the popularity of the artist as described in Section 2. However, if we change the definition of the morning, the user preferences might also change. Note, that these are the preferences that we want to predict and not the predictions.

In an off-line evaluation of the system, we compare the generated rating predictions to the hidden user ratings (hold-out evaluation), serving as a ground truth. But because our ground truth depends on the exact partitioning of time, intuitively we need to take into account all the possible partitions. Furthermore, we need a success measure in order to decide which partition is better. For this purpose, we propose to compute the error of the partition as the weighted average of all the errors in each segment:

$$E(R,T,D) = \frac{\sum_i |T_i| E(R,T_i,D)}{\sum_i |T_i|}$$

Where $D$ is the data set, $T = \{T_1, T_2, \ldots, T_t\}$ is the time partitioning of the time domain. Partitions do not overlap and the union of them is equal to $T$. $|T_i|$ is the number of the ratings we can predict in train set of the partition $T_i$. $E(R,T_i)$ is the Mean Absolute Error computed on the time partition $T_i$. The visual representation of possible partitioning is showed in Figure 3. Given the temporal partitioning $T$, the best system would be the one which makes smallest overall error $E$.

# 5. EXPERIMENTAL EVALUATION

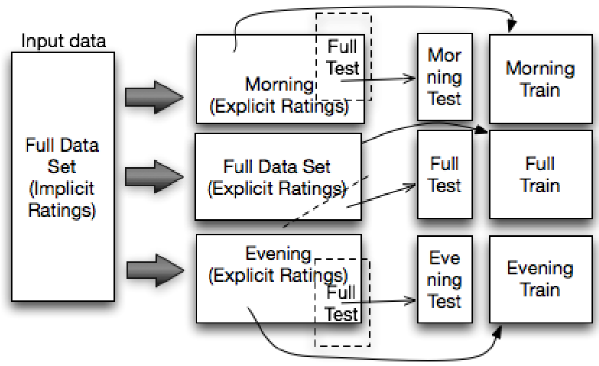



**Figure 4: Example of testing aproach for morning and evening partitions.**

For all our experiments we used last.fm data set described in Section 2. We used a popular factorization based CF algorithm (FACT for short)[1]. Our testing approach for the $T_{day}$ contextual segmentation is summarized in Figure 4. Due to the nature of implicit ratings, the procedure slightly differs from the usual off-line evaluation. In the first step, the implicit data set of the user is subdivided into contextual

[1] http://www.timelydevelopment.com

segments defined by $T_{day}$. Second, each of the segments is transformed into a $user \times item$ explicit rating matrix. We also transform the full data set into explicit ratings and divide it into the train and test sets. To be able to compare the performance of different contextual segments we use the same test set for each of the segment. The $user \times item$ pairs in the test set is used to extract the test set for each of the contextual segments. All $user \times item$ pairs that are present in the test set and contextual segment are extracted to the test set of that particular segment. The rest of the ratings are assigned to the train set of the same segment. Note that we can not split each segment into the train and test sets independently from each other because some ratings we are trying to predict could already be present in the training set of the segment. This procedure also allows us to use the training set of one segment to predict test set of other segment.

## 5.1 Accuracy of the Method

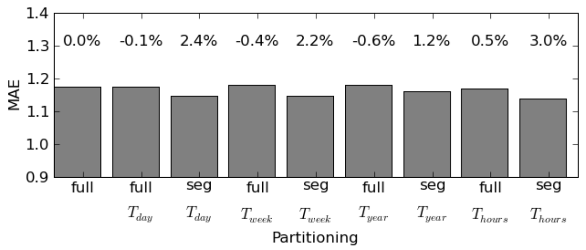



**Figure 5: Prediction accuracy for different segmentation.**

We shall now compare prediction accuracy of user micro-profiling and our baseline (context-free) prediction algorithm. We use a pre-defined time segmentation, which was done for the day, week and year temporal repetition. $T_{day} = \{morning, evening\}$, $T_{week} = \{weekend, working\ day\}$, $T_{year} = \{cold\ season, hot\ season\}$, $T_{hours} = \{even\ hours, odd\ hours\}$. Morning is defined as day hours between 5 am and 6 pm. Hot season includes spring and summer in Spain (March 21st to September 21st) . Even and odd hours is the partitioning that was used to test system behavior on the meaningless splits.

The goal of the experiment is to test if we can improve the accuracy $E$ of the predictor if we use only the profiles of the relevant segment. For example, for the day partitioning $T_{day}$ we use only the user micro-profile of morning to predict the ratings for the morning. We compare the prediction accuracy $E$ of this method to the prediction using the standard user profiles (without segmentation) to predict user preferences in the morning and in the evening. For all the experiments we use five fold cross-validation.

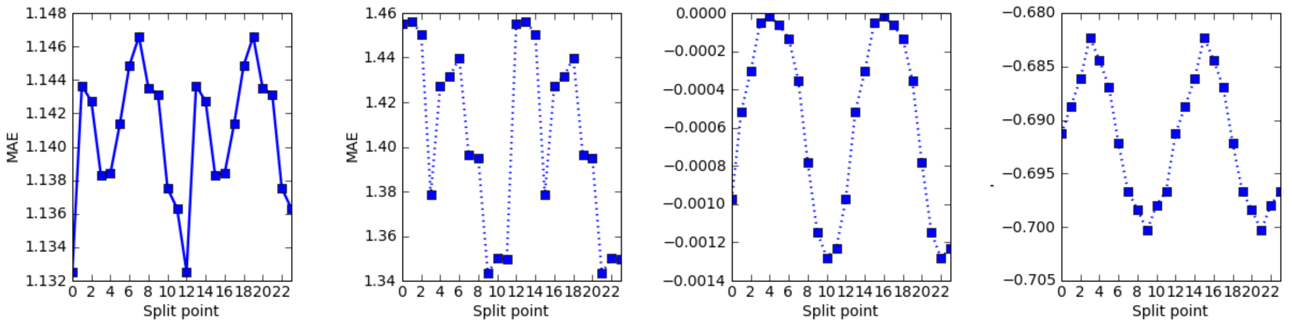
Figure 5 summarizes our results. The first column indicates the error $E$ of the FACT CF predictor using user profiles without segmentation and making the prediction for the full user profile (without segmentation). This column plays the role for the base line to which other results are compared. Following columns show the performance of algorithms to predict user preferences defined by the partitioning $T$. Algorithm makes prediction by using user profiles without partitioning (marked full on x axis) or using only the user micro-profiles that of the corresponding segment (marked seg on x axis). The experiment shows that prediction accuracy improves when using more relevant user micro-profiles. Note, that accuracy dropped when predicting user preferences for the contextual segment by using full user profile. It can be explained by the fact that to predict a specific (i.e., morning) user taste we use more general user data. On opposite, when using only the data of the segment the prediction improved significantly. We observe the highest improvement for $T_{day}$ and $T_{hours}$ partitioning. The improvement in $T_{hours}$ partitioning is unexpected and needs further analysis.

## 5.2 Similarities Between Splits

The previous experiment was conducted using pre-predefined time segmentations $T$. In our second experiment we aim at predicting the optimal split of the time variable. We examine the very simple case where the day cycle is partitioned into two segments each spanning for 12 hours. We want to find the optimal partition that reduces the overall error $E$. Figure 6 shows the true error $E$ and the methods used to predict this error. On the $y$ axis we plot the error (predicted error) $E$ and on the $x$ axis we plot the split point (i.e. every hour of the day). The graph is symetric with respect to the gap of the 12 hours. It is because our time segment is 12 hours itself and split at 0 o'clock is equal to the split at 12 o'clock. The true error is shown in the Figure 6(a). The error is computed every hour using the same prediction algorithm as in the first experiment. The minimum error in the day cycle is at 12 and 0 o'clock. This is surprising result as it corresponds with the start of the new day and the noon.

We use different methods to predict the true error $E$. Figure 6(b) shows the estimation of the true error $E$ using cross-validation. The cross-validation method is often used to estimate the free parameters of the algorithms. The split point could be seen as the parameter which needs to be optimizes with respect to the prediction error. We use 5 fold cross validation only on the train data – leaving out the test, that is – to compute the $E$. Figure shows, that the shape of the estimation resembles the shape of the true $E$. Note, that we try to predict only the optimal split point, therefore, are interested only in the minimum points of the error and not in predicting the absolute value of $E$. Cross-validation suggests that the minimum points are at 9 and 21 o'clock. The prediction is shifted to the left from the optimal solution by 3 hours.

However, using cross-validation to estimate the best split is expensive. It means running the recommendation algorithm several times for each possible split and this can be computationally unacceptable. Therefore, we compare this solution with two computationally cheaper methods. Both methods use proxy measures on the partitioned data set to compute the goodness of the split. The information gain (IG) method (see Figure 6(c)) uses this information theoretical measure to determine how much the split contributes to the knowledge of the data. Interestingly, the higher the information gain, the smaller is the error $E$. Analyzing the minimum point of the negative IG measure, we see that the minimum point is at 10 and 22 o'clock which is closer prediction comparing to the cross-validation method. The third method computes the mean explained variance of the first 100 principal components for the two data segments. Similarly to the IG method, the more variance is explained, the




(a) True error E (b) Cross-validation (c) Negative IG (d) Neg. Explained Variance

**Figure 6: Error and its estimation for different time splits.**

smaller is the error $E$. The minimum of the negative explained variance is at the 9 and 21 o'clock, which has the same prediction accuracy of the cross-validation method.

The experiment shows that predicting true error $E$ for even for the simple case can be a challenging tasks. Moreover, the well accepted cross-validation method can be outperformed by more lightweight heuristic approach such as computing IG for the split.

# 6. CONCLUSIONS AND FUTURE WORK

This work introduces and gives initial evaluation of the micro-profiling technique for time aware CF. We evaluated the method for different time splits and showed that using only the user micro-profile data for the prediction can increase the accuracy of the algorithm. We also present a novel evaluation technique for context-enriched implicit data. Moreover, we compared three different methods to find the optimal partition of the data. The experiments showed that the heuristic based methods can perform similarly good or outperform the more expensive cross-validation method.

In the future we plan to make more extensive evaluation of the micro-profiling approach. For initial evaluation we used user defined data splits. This has limitation as the possible splits are predefined and do not depend on data set and the user. We want to make more adaptive splits of the time domain. The split can be optimized for the whole data set or for each user separately. We expect to increase the accuracy of the current method. Moreover, we want to be able to combine the predictions made for different micro-profiles. For example, we could make user micro-profile for weekend and for the morning, and compute the predictions for both of them. When predicting a rating for morning on weekend we should combine both predictions. Here, the main challenge is to find the precise way to aggregate different recommendations.

In the initial experiments we made recommendations for the music albums but not the music tracks itself. We want to extend our approach and make recommendations on different levels of granularity, i.e., genre, artist, album and track. To our best knowledge this option has not been analyzed and could be very useful feature for exploratory recommendations. In our work, we use only time as the context of the user. We want to extend the context information to include current song and current album information.

Note, that all of these extensions are very related to the evaluation technique. Our currently proposed method does not allow overlapping time partitions. Moreover, the time partitions should be the same for all the users. In order to evaluate the method first we will have to find a meaningful way to compute the performance.

# 7. REFERENCES

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

[2] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, editors, *RecSys*, pages 335–336. ACM, 2008.

[3] M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. Mcgrath. Racofi: A rule-applying collaborative filtering system, 2003.

[4] O. Celma. Music recommendation and discovery in the long tail, 2008.

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

[6] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. *Data Mining, IEEE International Conference on*, 0:263–272, 2008.

[7] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

[8] O. Kwon and J. Kim. Concept lattices for visualizing and generating user profiles for context-aware service recommendations. *Expert Syst. Appl.*, 36(2):1893–1902, 2009.

[9] J. Lee and J. Lee. Context awareness by case-based reasoning in a music recommendation system. pages 45–58. 2008.

[10] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324. Springer Berlin / Heidelberg, 2007.