Matrix Factorization Techniques for Context Aware Recommendation

Linas Baltrunas
Free University of Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
lbaltrunas@unibz.it

Bernd Ludwig
Free University of Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
bernd.ludwig@unibz.it

Francesco Ricci
Free University of Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
fricci@unibz.it

ABSTRACT

Context aware recommender systems (CARS) adapt the recommendations to the specific situation in which the items will be consumed. In this paper we present a novel context-aware recommendation algorithm that extends Matrix Factorization. We model the interaction of the contextual factors with item ratings introducing additional model parameters. The performed experiments show that the proposed solution provides comparable results to the best, state of the art, and more complex approaches. The proposed solution has the advantage of smaller computational cost and provides the possibility to represent at different granularities the interaction between context and items. We have exploited the proposed model in two recommendation applications: places of interest and music.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

General Terms

Algorithms, Experimentation, Performance

Keywords

Recommender Systems, context-based reasoning, collaborative filtering, matrix factorization

1. INTRODUCTION

Recommender Systems (RS) are software tools helping online users to tame information overload by providing personalized recommendations on various types of products and services. Recently, Context-Aware Recommender Systems (CARS) have been proposed in order to address a limitation of classical RS: not taking into account the specific contextual situation in which the item will be consumed. Exploiting additional relevant information (context) CARS were shown to provide more accurate rating predictions [7, 10, 6] and more relevant recommendations [5].

CARS approaches can be divided into: contextual pre-filtering; post-filtering and contextual modeling [2]. The earlier research concentrated on pre-filtering approaches. For example, in the classical reduction based CACF approach [1], first the target user context is identified and consequently only the ratings previously collected in that context (or in a more general one) are used in generating a prediction. Actually, the reduction based approach requires the identification of the contextual data segments where it does improve the rating prediction over the baseline method that uses all the available data. Searching for these segments is expensive. Another pre-filtering approach called item-splitting [6] reduces the computational cost of reduction based on dynamically discovering, for each item, the relevant contextual factors. This method splits an item profile, i.e., the collection of its ratings, into two virtual items: if there is a statistically significant difference between the ratings for this item in two alternative contextual conditions. Item-splitting has been proved to provide more accurate rating predictions than reduction based. More recent CARS approaches propose to fit the rating data using regression models [7, 10]. Tensor Factorization (TF) is currently the most accurate model-based CARS technique [7]. TF extends the classical two-dimensional matrix factorization problem into an n-dimensional version of the same problem, which is called tensor factorization. The multi-dimensional matrix is factored into lower-dimensional representation, where the user, the item and each contextual dimension are represented with a lower dimensional feature vector.

Regression models, such as Tensor Factorization, introduce a huge number of model parameters that must be learned using the training data. In fact, in [7] it is shown that the number of model parameters grow exponentially with the number of contextual factors. While using large training data sets, TF was shown to improve the rating prediction accuracy of the heuristic based approaches mentioned above [7]. However, as we will show here when data sets are small, simpler models, with less parameters, can perform equally and even better. In fact, modeling components that are not useful to capture the dependency of the ratings from the context can have a negative impact on the rating prediction accuracy. For instance, when a contextual factor has no influence on the rating it plays the role of noise. Hence, we claim that CARS should consider the trade-off between model complexity and the amount of available training data,

and also the properties of the domain, i.e., the nature of the dependency between the contextual factors and ratings.

To explore these trade-offs in this paper we experimentally analyze three CARS models, which are much simpler than TF, and are based on Matrix Factorization. Each model makes different assumptions about the interaction of context and ratings. The simplest one, which has the smallest number of parameters, assumes that a contextual factor has a uniform influence on the ratings, irrespectively from the specific item considered. Conversely, the most complex one assumes that each contextual factor has different influence on the ratings for each item. In the middle, with respect to model complexity, we also propose a third one where we assume that the contextual factors have an equal impact on the ratings of all the items in a group, which is defined by a category feature (e.g., music with same genre). We use two real world data sets from the tourism and music domain and we show that this third model achieves a better generalization on the testing data.

In the next sections we first introduce the proposed models and then we present the evaluation results comparing these models with TF and the heuristic methods on two data sets that have been used in the previous literature. Then we report the results of the three proposed models on two new data sets showing that it can substantially improve the prediction accuracy of standard matrix factorization, i.e., when contextual information is not used. Finally we discuss the obtained results and we point to some future research work.

2. CONTEXT-AWARE FACTORIZATION

In this section, we present context-aware matrix factorization (CAMF): our extension of the classical Matrix Factorization (MF) approach for taking into account contextual information in the rating prediction step of a RS. CAMF generalizes an approach which was originally proposed in [9] to model the time dependency of ratings; but in addition CAMF can deal with a larger number of contextual factors. Moreover, in our approach we can deal with different granularities of the interaction of context with ratings. In particular, in this paper we discuss an instance of this general model that includes three levels of granularity.

In the first model, the most general one, we assume that each contextual condition has a global influence on the ratings independently from the item. For example, when the contextual factor "weather" is "sunny" one can detect that all the places of interest (POIs), independently from the nature of the particular POI, receive a higher rating. Hence, in this model, which is called **CAMF-C**, we introduce one single parameter for each contextual condition (i.e., value of a contextual factor). This parameter models how the rating deviates from the classical personalized prediction as the effect of the context: and this deviation is also called the *baseline* for that contextual condition.

In a second model we introduce one parameter per contextual condition and item pair. Clearly this has a finer grain, and it introduces a much larger number of parameters (to be learnt). This model can better predict ratings when the context influences differently the items. For instance, sunny weather could increase the rating of the flea market but not the rating for the museum of natural science. We call this model **CAMF-CI**.

In the third model, which has a middle complexity compared with the previous two, we introduce one model param-

eter for each contextual condition and item category. Here we are assuming that the items can be grouped in categories (provided by the domain expert), e.g., POIs can be museums or SPAs or bicycle paths, and music can be classified into genres. We call this model **CAMF-CC**.

Let us denote with r_{ui} the rating of user u for the item i, without any specific reference to the context in which this rating has been given. We will instead denote with r_{uic} the rating for i given by u in the contextual situation c, which is defined by the values of a set of contextual factors. In particular, a rating $r_{uic_1...c_k}$ provides the evaluation of the user u for the item i made in the context $c_1, \ldots, c_k$, where $c_j = 0, 1, \ldots, z_j$, and $c_j = 0$ means that the j-th contextual factor was unknown, while the other index values refer to possible values for the j-th contextual factor, i.e., are possible contextual conditions. The tuples $(u, i, c_1, \ldots, c_k)$, for which the rating $r_{uic_1...c_k}$ is known, are stored in the data set $R = \{(u, i, c_1, \ldots, c_k) | r_{uic_1...c_k} \text{ is known}\}$.

We recall that MF aims at factorizing the ratings matrix into two $m \times d$ and $n \times d$ dimensional matrices V and Q respectively. A user is then represented with a column vector $\vec{v}_u \in V$ and an item i with a column vector $\vec{q}_i \in Q$. In MF it is possible to balance the capacity and generalization capability of the predictive model by tuning the dimension d. Generalizing MF, we propose to model personalized context-dependent ratings with the following expression:

$$\hat{r}_{uic_1...c_k} = \vec{v}_u \cdot \vec{q}_i + \bar{\imath} + b_u + \sum_{j=1}^{k} B_{ijc_j} \qquad (1)$$

where $\vec{v}_u$ and $\vec{q}_i$ are d dimensional real valued vectors representing the user u and the item i. $\bar{\imath}$ is the average of the item i ratings in the data set R, b_u is the baseline parameter for user u. B_{ijc_j} are the parameters modeling the interaction of the contextual conditions and the items. Let us denote with $K = z_1 + \ldots z_k$, where k is the number of contextual factors and z_i is the number of possible values (conditions) of the i-th contextual factor. In CAMF-CI, the model with the finest grain, there is actually a parameter B_{ijc_j} for each contextual condition (of each contextual factor) and item i combinations. Hence if there are n items the total number of B_{ijc_j} parameters is Kn. The CAMF-CC model is coarser and there is one parameter for each contextual condition and items' category pair. In practice, if the items i and f have the same category, then $B_{ijc_j} = B_{fjc_j}$, and if t is the number of different categories (in our data sets we have 5 and 10 categories) the total number of parameters is Kt. Finally, in CAMF-C there is just one single parameter B_{jc_j} for each contextual condition (value of a contextual factor), i.e., $B_{ijc_j} = B_{fjc_j}$ for every item i and f. We note that, in general, if a contextual condition is unknown, i.e., $c_j = 0$ then the corresponding baseline parameter B_{ijc_j} is set to 0.

We observe that the proposed model could be extended to take into account the possible dependencies between contextual factors. One could add additional parameters $B_{ijc_jlc_l}$, modeling the dependency between the conditions c_j and c_l of the j-th and l-th factor. This more complex model could better fit the data but, if there are not enough training data the increased complexity can have a negative effect on model accuracy. In this paper we deal with small data sets and therefore we have opted for simpler models and not considering the interaction of contextual factors. Similarly, one could model the relationship between context and users.

However, such an extension even if could result in a better rating prediction it can not change the ranking of the items for a user.

In order to generate rating predictions, the model parameters should be learned using the training data. We define the learning procedure as an optimization problem:

$$\min_{v_*, q_*, b_*, B_*} \sum_{r \in R} \left[\left(r_{uic_1 \ldots c_k} - \vec{v}_u \cdot \vec{q}_i - \bar{\imath} - b_u - \sum_{j=1}^{k} B_{ijc_j} \right)^2 \right.$$
$$\left. + \lambda \left(b_u^2 + \|\vec{v}_u\|^2 + \|\vec{q}_i\|^2 + \sum_{j=1}^{k} \sum_{c_j=1}^{z_j} B_{ijc_j}^2 \right) \right]$$

where $r = (u, i, c_1, \ldots, c_k)$ and R is the set of context-dependent training ratings. For better generalization performance, a regularization term is added, as it is usual in this type of models. Regularization is controlled by the λ meta parameter. As λ grows the model becomes more "rigid", and fits less the training data. We have used stochastic gradient descent for solving this problem. This has been proved to be an efficient approach [8]. This procedure updates one after another all the parameters, moving them in the opposite direction of the gradient, while all the other parameters are kept unchanged. The learning rate depends on the meta parameter γ; in our experiments this was set to 0.001.

The proposed models can be trained in linear time with respect to the number of data points and contextual factors. This is a big advantage as the algorithm can be used with many contextual factors. We tested it with up to 14 contextual factors (dimensions) containing in total 52 contextual conditions. Note that, the training and prediction step of the state of the art CACF method base on Tensor Factorization is exponential in the number of contextual dimensions.

3. EXPERIMENTAL EVALUATION

We have evaluated the proposed methods on real world and synthetic data sets. The summary descriptions of the data sets are given in the Table 2. We will first show that CAMF has comparable performances to current best CARS methods, i.e., those based on Tensor Factorization (TF) [7]. Later we will provide a more detailed analysis of CAMF on two real world data sets. We have compared CAMF with TF on three semi-artificial data sets and one real world data set. The semi-artificial data sets were generated using the Yahoo Webscope movie data [11]. These data sets contain an artificial feature that simulates a contextual factor, the higher the α value, the more influential is this factor. We refer to [6] for further details on these data sets. We have also used one real world data set (MovieAT) that was firstly used in [1]. For our comparisons here we have used the CAMF-CI method. We did not use CAMF-CC and CAMF-C as in the Yahoo! derived data sets the artificial feature by construction does not depend on the item category.

We have estimated the performance of the considered models using repeated random sub-sampling validation. We have generated 25 splits into training and testing set with the training set containing 90% of the data. We have computed the Mean Absolute Error (MAE) of each method. The summary of the results is shown in Table 1. CAMF performs better than TF on the small real world data set (MovieAT) and on the semi-artificial data sets where context has small influence $Yahoo!_{\alpha=0.1, \beta=0.9}$. The better per-

Table 1: MAE of TF and CAMF

Data Set	CAMF-CI	TF
MovieAT	2.04	2.10
$Yahoo!_{\alpha=0.1, \beta=0.9}$	0.748	0.758
$Yahoo!_{\alpha=0.5, \beta=0.9}$	0.750	0.689
$Yahoo!_{\alpha=0.9, \beta=0.9}$	0.672	0.649

formance on MovieAT data can be explained by the general rule that for small data sets a simpler model, i.e., with less parameters generally performs better. Moreover, CAMF uses item and user baselines and when the influence of the context is light, as in $Yahoo!_{\alpha=0.1, \beta=0.9}$ this simple model component contributes to produce a smaller generalization error than TF. When context has a stronger influence, e.g., in $Yahoo!_{\alpha=0.5, \beta=0.9}$ and $Yahoo!_{\alpha=0.9, \beta=0.9}$, and there are enough data to learn the TF model parameters, then TF outperforms CAMF. We want also to mention that the performance of both (model-based) TF and CAMF on these data sets are better than the reduction based approach [1] and item-splitting [6], which are pre-filtering techniques (not shown here for lack of space).

To better understand the relationship between model complexity and context we conducted another set of experiments where we used two real world data sets from a tourism and music recommendation applications. These data set were collected using two web interfaces where users were asked to imagine some specific contextual conditions to hold and then to rate a place of interest or a music track. The details of the data collection procedure are illustrated in [5] and [4] and the summary is given in Table 2.

Figure 1 shows the MAE of the considered models. Here we compare CAMF with standard MF (without context) and with the simple model that predicts a rating as the item average rating (denoted AVG in all the figures). The model complexity, which is measured by the number of model parameters, increases from left to right. For example, AVG is the simplest one and CAMF-CI is the most complex. In particular, in the tourism domain data set: AVG uses 21 model parameters (the items' averages); MF has 282 parameters; CAMF-C has 335 parameters; CAMF-CC has 547 parameters; and CAMF-CI has 1395 parameters. The largest improvement with respect to AVG is achieved, as expected, by personalizing the recommendations, i.e., using the classical MF. This gives an improvement of 11.3% on the tourism domain data and 19.1% on the music domain data. These improvements are similar to those reported in other studies that have compared personalized vs. non-personalized rating predictions (e.g., [3]). But, the personalized model can be further improved by contextualization. All the three context-aware MF models produce an improvement ranging from 18.7% to 21.6% on the tourism domain data and from 19.2% to 22.6% on the music domain data, compared with AVG.

Interestingly, as we observed when commenting the results of the previous experiments, the best performing context-aware MF method is not the most complex one. In fact, CAMF-CC outperforms the other methods on both data sets. Increasing the model complexity, using CAMF-CI, reduces the rating prediction accuracy.

Table 2: Summary of the data sets

Data set	ratings	users	items	item categories	contextual factors	contextual conditions
Tourism	1679	26	21	5	14	53
Music	4013	44	140	10	8	27
MovieAT	1464	84	192	-	4	16
$Yahoo!_{\alpha=*,\beta=*}$	221K	7.6K	11.9K	-	2	4

(a) Tourism   (b) Music
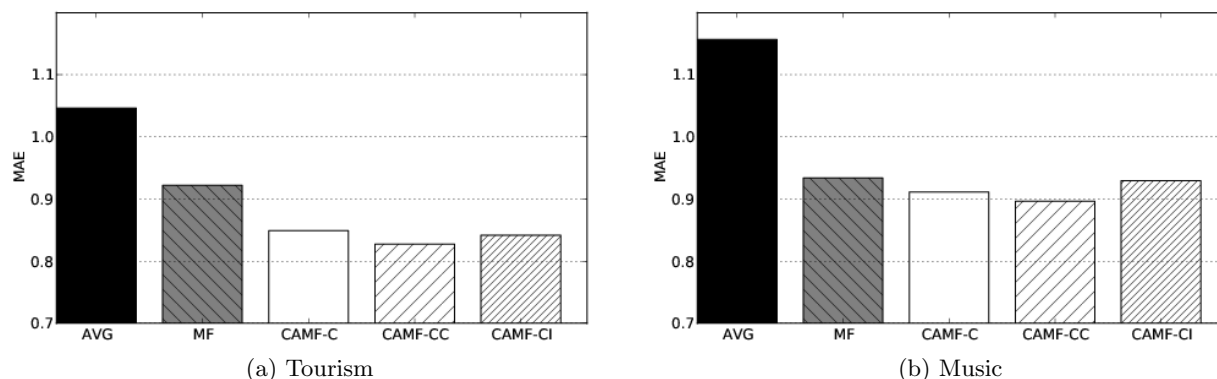
Figure 1: Mean Absolute Error of the compared methods

4. DISCUSSION AND CONCLUSIONS

In conclusion, we can state that the proposed modeling approach can substantially improve the rating prediction accuracy by taking into account contextual information. Moreover, our results indicate that the best model granularity, with respect to the interaction of context and items, depends on the domain and amount of data available. On the available data sets, grouping the items per item category has shown to have a beneficial effect.

We must observe that we still need to perform a more comprehensive experimental evaluation of the proposed approach. In fact, we stress that context-awareness could be beneficial only to some extent, and it is important to estimate the sensibility of the data to different granularities of the context model. In that respect we plan to perform a more extensive comparison of TF and CAMF.

5. REFERENCES

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

[2] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 217 – 250. Springer, 2011.

[3] M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. McGrath. Racofi: A rule-applying collaborative filtering system. In *Proceedings of COLA03*. IEEE/WIC, October 2003.

[4] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Luke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *Proceedings of the 12th International Conference on Electronic Commerce and Web Technologies*, pages 89–100, Toulouse, France, 2011.

[5] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, pages 1–20, 2011.

[6] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 245–248, New York, NY, USA, 2009. ACM.

[7] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86, New York, NY, USA, 2010. ACM.

[8] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 426–434, New York, USA, 2008. ACM.

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

[10] Y. Shi, M. Larson, and A. Hanjalic. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Challenge on Context-aware Movie Recommendation*, 2010.

[11] Yahoo! Research Webscope Movie Data Set. http://research.yahoo.com/, Version 1.0. Received in, 2007.