Using Graded Implicit Feedback for Bayesian Personalized Ranking

Lukas Lerche
TU Dortmund, Germany
lukas.lerche@tu-dortmund.de

Dietmar Jannach
TU Dortmund, Germany
dietmar.jannach@tu-dortmund.de

ABSTRACT

In many application domains of recommender systems, explicit rating information is sparse or non-existent. The preferences of the current user have therefore to be approximated by interpreting his or her behavior, i.e., the implicit user feedback. In the literature, a number of algorithm proposals have been made that rely solely on such implicit feedback, among them Bayesian Personalized Ranking (BPR).

In the BPR approach, pairwise comparisons between the items are made in the training phase and an item i is considered to be preferred over item j if the user interacted in some form with i but not with j. In real-world applications, however, implicit feedback is not necessarily limited to such binary decisions as there are, e.g., different types of user actions like item views, cart or purchase actions and there can exist several actions for an item over time.

In this paper we show how BPR can be extended to deal with such more fine-granular, graded preference relations. An empirical analysis shows that this extension can help to measurably increase the predictive accuracy of BPR on realistic e-commerce datasets.

Categories and Subject Descriptors

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

Keywords

Recommender Systems; Implicit Feedback; Evaluation

1. INTRODUCTION

Most of the published Collaborative Filtering (CF) recommendation approaches aim to identify patterns of user preferences within a set of *explicit item ratings* provided by a larger user community. In practice, however, this form of explicit feedback from the users in many application domains is very sparse, limited to dominantly positive or positive-only statements, or even non-existent.

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

http://dx.doi.org/10.1145/2645710.2645759.

Correspondingly, a number of proposals have been made in the literature to exploit *implicit user feedback* in the recommendation process, e.g., [3, 7, 8, 10]. In these cases, a user's true preferences are estimated by interpreting his or her behavior. On an online shop, for example, one will typically interpret user actions like an item purchase or viewing an item for some time as a sign of interest in the item.

Bayesian Personalized Ranking (BPR) [10] is a comparably recent CF method designed to deal with implicit-only feedback. One particularity of BPR when compared, e.g., to the highly accurate SVD++-based method [3] is that it directly aims to optimize a ranking criterion, whereas SVD++ primarily aims to minimize the prediction error. Therefore, BPR can be considered to fall into the class of "learning-to-rank" methods [2] which – due to their design – are typically able to outperform methods designed for error-minimization on *rank measures* like precision, recall or nDCG.

In the BPR approach, the input consists of a user-item rating matrix, which however only captures positive interactions (unary feedback), i.e., there is a "one" whenever a user i has interacted with item j. At its core, BPR uses the pairwise preferences derived from the implicit ratings to optimize an AUC-like ranking criterion and Rendle et al. show how this optimization can be applied to two typical classes of recommendation models (kNN and matrix factorization).

One limitation of the basic BPR model is that it only supports discrete, unary feedback. In practice, however, some types of user interactions might be considered stronger indicators of interest than others. A purchase action might be stronger than a view action; multiple clicks on an item might be more indicative than a single click; finally, a recent view action might be more relevant than one in the past.

In this work, we propose an extension to BPR (called `BPR++`), which is designed to take various forms of such graded implicit feedback into account. The main idea is to first derive additional pairwise preferences from the data using the available "importance" information and – in the optimization phase – bias the optimization procedure to draw a certain amount of samples from these additionally available data points. In the next section, we describe the approach in more detail before we present results of an experimental evaluation with different datasets.

2. EXTENDING BPR

2.1 Generation of preference relations

In BPR, the first task is to derive pairwise item preferences from the unary feedback. The set of available items is denoted as I. For each user u, we are given a set of items I_u^+, containing those items of I for which a (positive) implicit feedback action was observed. User-specific item preferences are triples of the form (u, i, j), which express the fact that user u prefers item i over item j. The set of derived preference relations D_S, i.e., the training data used for optimization later on, is defined as follows.

$$D_S := \{(u, i, j) \mid i \in I_u^+ \wedge j \in I \backslash I_u^+\}$$

Using this approach, we can encode single, positive-only statements like non-repeated item purchases or "like" statements on social platforms. The user actions that can be used to assess preference profiles can however be richer in reality and may include, for example, the following:

1. In the logs of online shops, we typically can find view, add-to-cart, add-to-wishlist, or purchase actions, which can correspond to interest indicators of different strengths. In an even more fine-grained setting, the individual viewing times could be taken into account.
2. Recency of events and temporal aspects could be considered as carrying information about (recent) user preferences.
3. There are specific relationships between the users, e.g., in the form of social network connections [5], or between items that share similar characteristics.

In this paper, we therefore propose a generic extension of BPR regarding the way the set D_S is derived. In the original BPR formulation, a preference can only be derived if there is an implicit rating for i and none of j (e.g., because the user only viewed item i). If, for example, the user viewed i three times and j one time, nothing can be said about the preference. Since the number of views can be an interest indicator, we introduce a preference weight function $pweight(u, i)$ for user actions instead of the binary decision of [10].

We assume that $pweight(u, i)$ returns a positive number if an interaction between u and i exists, and 0 otherwise. A preference tuple (u, i, j) is therefore inferred whenever $pweight(u, i) > pweight(u, j)$ holds. Generally, the *pweight* function can encode arbitrary types of information, including, e.g., the time or recency aspects as mentioned above. The extended set of preference relations D_S^{++} for each user u is defined as follows:

$$D_S^{++} := \{(u, i, j) | pweight(u, i) > pweight(u, j), i \in I, j \in I\}$$

Consider the example in Figure 1. Let us assume that we have information about how often a user viewed an item as well as the time point of the latest view action for each item. User u_1 has both viewed item i_1 and item i_2. In the original BPR formulation, no preference could be derived. Here, however we could design a function *pweight* that returns a higher value for i_1 because of the higher number of interactions. Likewise, the function could return a slightly higher preference value for item i_1 also for user u_2 because of the more recent view event. A weighted combination of these aspects inside *pweight* is also possible. For user u_3, both the original and adapted BPR version would derive a preference for i_1. Generally, how *pweight* is defined depends on the application domain and the characteristics of the available data. Meta-data features or social network information could for example be used as done, e.g., in [5] or [7]. The chosen *pweight* function in any case however has to be carefully designed and thoroughly evaluated.

User \ Item	i_1	i_2
u_1	2 (t=10)	1 (t=10)
u_2	1 (t=12)	1 (t=10)
u_3	1 (t=12)	?

Table 1: Implicit feedback database.

2.2 Biased sampling

In the learning phase of BPR, Rendle et al. proposed to use a random bootstrap stochastic gradient descent approach to choose triples from D_S to learn the model, since using all $O(|U| \cdot |I|^2)$ triples is in general not feasible. When using the same random sampling in the `BPR++` approach, most of the time only triples from the original BPR formulation would however be considered. The reason is that the number of additional data tuples (u, i, j) in D_S^{++} can be fairly small when compared to the number of triples in D_S, because additional tuples are only generated when the user has interacted both with the items i and j.

We therefore extend the training algorithm of `BPR++` with a parameter β that expresses the probability of taking a sample from the additional tuples in D_S^{++}. Finding a suitable value for β is crucial for the effectiveness of the approach and its value has to be determined empirically for a given data set. The modification to the learning procedure of [10] is shown in lines 4-6 in Figure 1.

1: **procedure** LearnBPR++(D_S,Θ)
2: initialize Θ
3: **repeat**
4: $\quad r$ = random(0,1)
5: $\quad$ **if** $r \leqslant \beta \wedge D_S^{++} \backslash D_S \neq \varnothing$ **then**
6: $\quad\quad$ draw (u,i,j) from $D_S^{++} \backslash D_S$
7: $\quad$ **else**
8: $\quad\quad$ draw (u,i,j) from D_S
9: $\quad$ **end if**
10: $\quad \Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} - \lambda_\Theta \cdot \Theta \right)$
11: **until** convergence
12: **return** $\hat{\Theta}$
13: **end procedure**

Figure 1: Learning algorithm with biased sampling.

3. EXPERIMENTAL EVALUATION

To determine to which extent the consideration of different types of feedback helps to improve the prediction accuracy of BPR, we conducted experiments on different data sets and, correspondingly, different types of additional information. The accuracy was measured in terms of precision and recall[1]. We used two datasets consisting of navigation log data from real online shops as well as one of the MovieLens datasets. Table 2 shows the basic statistics of the datasets.

Overview of datasets.

Zalando: A dataset from Zalando[2], a large online retailer for fashion products, that has been sampled from log data in

[1]Additional MRR measurements were in line with these results for all datasets and are therefore not reported here.
[2]http://www.zalando.de

such a way that no conclusions on visitor data or true business figures of the company can be drawn. The logged and session-numbered interactions have different types (view, purchase, wish, put-in-cart), where the majority of the events are item views. We created a subset for which there are at least 10 interactions per user and item.

Tmall: A similar but smaller anonymized dataset from a shopping portal[3] containing user interactions of different types. For each of the actions, a time stamp is available.

MovieLens: We created a subset of "heavy users" of MovieLens using the 10 million ratings dataset as a basis.

Dataset	Zalando	Tmall	MovieLens
users	12,724	884	17,107
items	20,596	9,531	963
interactions	2,021,716	182,879	3,601,968
sparsity	0.008	0.022	0.219
interactions/user	159	207	211
interactions/item	98	19	3,740

Table 2: Dataset characteristics.

BPR++ variants and measurement method.

Depending on the available data, we used the following comparably simple variants of `BPR++` and varied the interpretation of $pweight(u, i)$ correspondingly.

`BPR++(T)`: Items with which the user interacted more recently are considered to be more relevant. $pweight(u, i)$ corresponds to the time stamp or session number of the last interaction of user u with item i.

`BPR++(N)`: Items with which the user interacted more often are considered to be preferred. $pweight(u, i)$ corresponds to the number of interactions of user u with item i.

The time-based and interaction-count based versions were tested for the shop datasets. For the MovieLens dataset, we used the rating values as a preference indicator.

`BPR++(R)`: Higher rated items are more relevant. $pweight(u, i)$ corresponds to the rating of user u for item i.

In our evaluation, we compared `BPR++` with the original BPR implementation[4], FunkSVD[5] with 50 latent features, and item-to-item collaborative filtering with the cosine similarity measure (Item-KNN). The number of optimization rounds for the BPR-based methods was varied in the tests to analyze the convergence and possible overfitting behavior of `BPR++`. We have empirically evaluated different values for β and determined a fixed value $\beta = 0.75$ in all our experiments as it led to overall high results and enables comparability between the different $pweight(u, i)$ functions[6].

For the online shop datasets we created one time-based 80/20 train-test split. Since cross-validation is not possible due to the unique time-based ordering, we randomly created ten 90% sub-samples of the splits and repeated the experiments to factor out random effects. For the MovieLens dataset we used standard 5-fold cross-validation.

[3]Tmall Rec. Prize 2014 & TianChi Open Data Project

[4]We used matrix factorization, uniform sampling, 100 latent features, $\alpha = 0.05$, $\lambda_W = \lambda_{H^+} = 0.0025$, $\lambda_{H^-} = 0.00025$

[5]http://sifter.org/~simon/journal/20061211.html

[6]Optimizing β for the different datasets and $pweight(u, i)$ functions is beyond the scope of this paper.

Precision@10 and Recall@10 were used as accuracy measures. An item was considered relevant when it was actually purchased (shop data) or rated higher than the user's average (MovieLens). Items that were previously bought or rated were not recommended for the Zalando and movie dataset. Since in the *Tmall* scenario rather families of items than individual items are recommended, we allowed "repeated recommendations" for this dataset.

4. RESULTS

Zalando: Table 3 shows precision and recall for the Zalando dataset after n optimization rounds[7]. The time-based variant `BPR++(T)` helps to increase both precision and recall which indicates that the recency of the interactions is relevant in the domain. The `BPR++(N)` variant in contrast led to results that were worse than the original BPR method, which is possibly caused by recency effects. Since `BPR++(N)` ignores the interaction time, the focus on items that are no longer relevant for the user might be too strong. Systematic tuning of β might help to mitigate this effect.

#it	`BPR++(T)`	`BPR++(N)`	BPR
1	0.005/0.024	0.004/0.020	0.002/0.009
10	0.044/0.193	0.025/0.113	0.033/0.149
50	0.064/0.268	0.037/0.163	0.048/0.209
100	0.068/0.285	0.041/0.177	0.051/0.221
150	0.070/0.289	0.042/0.183	0.054/0.230
200	**0.071/0.293**	0.043/0.186	0.054/0.231

Table 3: Precision@10/Recall@10 for *Zalando*

Tmall: The results for the *Tmall* shown in Table 4 corroborate the findings made for the Zalando dataset. Taking recency of events into account (`BPR++(T)`) helps to improve the accuracy both in terms of precision and recall. Considering the number of interactions for each item as preference indicators (`BPR++(N)`) leads to faster convergence – which can be considered a highly desirable characteristic for large datasets – but not to better overall results. Again, Funk-SVD, item-kNN, and popularity-based ranking performed worse.

#it	`BPR++(T)`	`BPR++(N)`	BPR
1	0.041/0.252	0.030/0.183	0.020/0.118
10	0.075/0.457	0.051/0.314	0.045/0.277
50	0.085/0.513	0.056/0.343	0.056/0.341
100	0.087/0.521	0.057/0.344	0.056/0.341
150	**0.088/0.522**	0.058/0.352	0.059/0.362
200	0.087/0.521	0.058/0.350	0.059/0.360

Table 4: Precision@10/Recall@10 for *Tmall*

MovieLens: The results for the MovieLens data are shown in Table 5. The rating-based variant of `BPR++` performed best already after only 10 iterations. With an increasing number of iterations, there is however a tendency of overfitting[8].

[7]Results for Funk-SVD, item-kNN, and popularity-based ranking are omitted, since they never exceeded the value 0.001 for precision and recall.

[8]The results obtained for the original BPR are comparable to the results reported at `http://mymedialite.net/examples/item_recommendation_datasets.html`. item-kNN and popularity-based ranking led to lower results than FunkSVD and are therefore again omitted here.

#it	BPR++(R)	BPR	FunkSVD
1	0.368/0.171	0.331/0.153	0.092/0.040
10	**0.398/0.191**	0.354/0.169	0.212/0.096
50	0.387/0.186	0.341/0.163	0.164/0.072
100	0.387/0.186	0.336/0.161	0.145/0.061
150	0.382/0.183	0.342/0.164	0.144/0.060
200	0.384/0.184	0.342/0.165	0.140/0.058

Table 5: Precision@10/Recall@10 for MovieLens

We made additional experiments in which we used the non-filtered MovieLens dataset. In these scenarios, the results of BPR++(R) were however worse than when the original BPR method was used. Overall, our results show that additional preference indicators and graded forms of implicit feedback can help to increase the prediction accuracy of BPR[9]. The choice of the additional preference signals however depends on the domain. Using frequency information as an indicator for preference strength did for example not lead to better results in the Zalando scenario but actually led to a deterioration of the results.

So far, we have only considered one additional factor at a time to derive pairwise item preferences. On principle, multiple types of information could be incorporated in parallel, which should help us to achieve a higher density of the preference matrix. One particular question here is how to deal with situations when the preference statements that were derived from different information sources are conflicting. The design and evaluation of such hybrid methods and the analysis of factors like diversity and potential popularity-biases are part of our ongoing work.

5. RELATED WORK

BPR falls into the category of "one-class collaborative filtering" approaches and is able to make inferences from positive-only feedback (interpreted user actions). At the same time, with the AUC-like optimization criterion, BPR can be considered as a "learning-to-rank" approach. Another learning-to-rang technique is xCLiMF [11] which uses a graded relevance scale and optimizes a rank criterion, in that case the Expected Reciprocal Rank.

In recent years, a number of techniques have been developed to better deal with implicit user feedback, among them the SVD++ method, which integrates both implicit and explicit feedback in a matrix factorization model [3]. In [7], Manzato shows how to integrate additional information about item content into SVD++. In [4], the SVD++ model is extended to be able to deal with temporal dynamics. In our work, we have shown how time and recency information can be incorporated into the BPR method. In contrast to other time-aware CF methods, considering temporal aspects is however only one possible way of incorporating graded relevance feedback. Using interaction counts on items was explored as another possible form of graded feedback in our paper. This is partially similar to the probabilistic method presented in [6], which can use interaction frequency information when learning a preference profile.

Extensions to the basic BPR scheme have for example been proposed in [1], [5], or [9]. In [1], a strategy that weights the impact of negative items based their global popularity is introduced. In [5], the authors integrate additional information – in that case social network information – into BPR. Different to our work, their method directly influences the weight learning approach. In [9], an alternative sampling strategy was proposed that aims to statically or adaptively determine the most informative pairs.

[9]The differences of the best BPR++ results and the BPR results were statistically significant (p<0.01) for all datasets.

6. CONCLUSIONS AND OUTLOOK

In this work we propose to incorporate implicit feedback signals at a more fine-grained level in the personalized ranking process. The approach therefore allows us to differentiate between arbitrary types of feedback, e.g., take time and recency aspects into account. An experimental evaluation shows that for the considered domains the retrieval accuracy can be improved and faster convergence can be achieved when compared to the original BPR approach. In our future work, we plan to explore additional ways of combining the different types of user feedback in a hybrid approach, further investigate temporal effects and also look at other characteristics of the generated recommendation lists such as diversity and popularity biases.

7. REFERENCES

[1] Z. Gantner, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Bayesian personalized ranking for non-uniformly sampled items. *JMLR*, 18:231–247, 2012.

[2] A. Karatzoglou, L. Baltrunas, and Y. Shi. Learning to rank for recommender systems. In *Proc. RecSys '13 (Tutorials)*, pages 493–494, 2013.

[3] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. KDD '08*, pages 426–434, 2008.

[4] Y. Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4):89–97, 2010.

[5] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proc. WSDM '12*, pages 173–182, 2012.

[6] H. Ma, C. Liu, I. King, and M. R. Lyu. Probabilistic factor models for web site recommendation. In *Proc. SIGIR '11*, pages 265–274, 2011.

[7] M. G. Manzato. gSVD++: Supporting implicit feedback on recommender systems with metadata awareness. In *Proc. SAC '13*, pages 908–913, 2013.

[8] D. Oard and J. Kim. Implicit feedback for recommender systems. In *Proc. AAAI'98 Workshop on RS*, pages 81–83, 1998.

[9] S. Rendle and C. Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proc. WSDM '14*, pages 273–282, 2014.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. UAI '09*, pages 452–461, 2009.

[11] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, and A. Hanjalic. xCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance. In *Proc. RecSys '13*, pages 431–434, 2013.