Alpha-Beta Sampling for Pairwise Ranking in One-Class Collaborative Filtering

Mingyue Cheng[†], Runlong Yu[†], Qi Liu[†*], Vincent W. Zheng[‡], Hongke Zhao[§], Hefu Zhang[†], Enhong Chen[†]
[†]Anhui Province Key Laboratory of Big Data Analysis and Application, University of S&T of China, Hefei, China
{sa517038, yrunl, zhf2011}@mail.ustc.edu.cn, {qiliuql,cheneh}@ustc.edu.cn
[§]Tianjin University, Tianjin, China, {hongke@tju.edu.cn}
[‡]WeBank, Shenzhen, China, {vincentz@webank.com}

***Abstract*—This paper introduces Alpha-Beta Sampling (AB-S) strategy, which is particularly intended for the sampling problem of pairwise ranking in one-class collaborative filtering (PROCCF). Specifically, ABS strategy places more emphasis on such training examples, including positive item with a lower preference score and negative items with a higher preference score for each gradient step. Then, we provide the corresponding proofs for the ABS strategy from both gradient and ranking perspectives. First, we prove that sampled training examples by ABS strategy can update the model parameters with a large magnitude and analyze two instantiations by combining two specific pairwise algorithms. Second, it can be proved that ABS strategy is equivalent to optimizing for ranking-aware evaluation metrics like Normalized Discounted Cumulative Gain (NDCG). Furthermore, ABS strategy can be very general and applicable in a lot of pairwise structures of pairwise algorithms. Based on ABS strategy, we provide an effective sampling algorithm to dynamically draw items for each SGD update. Finally, we evaluate the ABS strategy by conducting sampling tasks in two representative pairwise algorithms. The experiment results show that the ABS strategy performs significantly better than the baseline strategies.**

***Index Terms*—Sampling, Pairwise Preference Learning, One-Class Collaborative Filtering**

I. Introduction

As one of the most popular and effective recommending methods [1], [2], Collaborative Filtering (CF) [3] has become the core technologies of recommendation systems. It makes the recommendation by leveraging the user-item preference patterns derived from a large amount of historical data. Different from explicit feedback [4] which can explicitly express users' preference with rating scores, one-class (implicit feedback [5]) recommendation problems lack negative feedback. To tackle the one-class recommendation problems, previous literatures pointed out that pairwise methods have become the preferred solutions than other methods. Due to the large number of item pairs, a common practice is to employ Stochastic Gradient Descent (SGD) to optimize pairwise algorithms. For each gradient step, it draws a positive item from the observed item set and selects a negative item from the unobserved item set to construct training instances. Such a process is also known as the sampling process.

Uniform sampler has been widely adopted by a lot of pairwise methods. Specifically, uniform sampler randomly draws item pairs to make up training instances, and does not distinguish the candidate ones. However, it is shown that uniform sampler might draw some meaningless training examples which can make no effect on the gradient update. To solve the ineffectiveness problem in pairwise preference learning, some non-uniform sampling strategies [6], [7] were proposed by sampling negative items in a fine-grained way. Although these strategies have improved the effectiveness of pairwise preference learning, there are several limitations in the existing works. First, previous works do not utilize both positive and negative item at the same time. Second, previous works ignore that there exist various pairwise structures in different pairwise algorithms. Hence, how to design a fine-grained strategy, which can utilize the rank information of both positive and negative items, is crucial to solving the ineffectiveness problem in pairwise preference learning.

In this paper, we focus on studying the problem of sampling tasks for pairwise ranking in one-class collaborative filtering (PROCCF). To address these aforementioned problems, we proposed Alpha-Beta Sampling (ABS), a novel sampling strategy by drawing such examples for which the model currently predicts very incorrectly. Specifically, ABS strategy place more emphasis on such training instances, including positive item with lower preference score and negative one with higher preference score, for each gradient step. Along this line, we provide the corresponding proofs for ABS strategy why can work well for pairwise preference learning from two perspectives: gradient update and ranking task. Furthermore, we claim that the proposed ABS strategy can be very general and applicable to different pairwise structures in various pairwise algorithms. Based on ABS strategy, an alternative sampling algorithm is developed to dynamically select training instances for each SGD update. Finally, we perform thorough experiments on three real-world datasets and compare ABS strategy with state-of-the-art strategies. Our results show that ABS strategy outperforms all baselines in terms of convergence and ranking performance.

II. Preliminaries

A. Notations

Suppose that $U = \{u\}_{u=1}^{n}$ denotes a set of users and $I = \{i\}_{i=1}^{m}$ is defined as the set of items, where n and m respectively represent the number of users and items. $D = \{(u, i)\}$ denotes a set of implicit feedbacks. I_u^+ represents the set of items that user u have given positive feedback, and

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00114

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.

$I \backslash I_u^+$ denotes the unobserved item set. Generally, by making full use of implicit feedback $\{j|(u,j) \notin D\}$ for each user u, the goal of recommendation methods is to find a rank function, which can generate a personalized ranked list of items from the unobserved item set (i.e., $I \backslash I_u^+$) by picking up such items which are the most relevant to the user. In order to represent user' preference over items with only implicit observations R, pairwise preference learning approaches typically regard observed user-item pair $(u,i) \in I_u^+$ as a positive label, and all other combination $(u,j) \in I \backslash I_u^+$ as negative feedback.

B. Pairwise Preference Learning

Generally, pairwise approaches take pairs of items as basic units to maximize the likelihood of pairwise preferences over observed items and unobserved ones. Particularly, there is a fundamental assumption widely adopted in pairwise methods, and it can be described as follows:

Assumption 1 *User u prefers item i than item j, $(u,i) \succ (u,j)$, if item i is sampled from the observed item set and negative item j is sampled from the unobserved ones*

With sampled user-item set (u,S), pairwise algorithm usually minimizes the tentative objective function $f(u,S)$. In order to encourage pairwise competition, the tentative objective function usually can be defined as:

$$f(u,S) = f(\succ_u), \tag{1}$$

where $\succ_u$ denotes the pairwise preference structure for a given user u. For a specific model, e.g., in BPR where $r_{\succ_u}$ denotes $r_{ui} > r_{uj}$, which is the difference between user u's preference on item i and item j, and $S = \{i,j\}$. For a given user u, the preference, i.e., r_{ui}, between user u and item i can be modeled by a set of parameters denoted by θ, which include user u's latent feature vector $U_{u\cdot} \in \mathbb{R}^{1\times k}$, item i's latent feature vector $V_{i\cdot} \in \mathbb{R}^{1\times k}$ and item i's bias $b_i \in \mathbb{R}$. Note that k is the number of latent vectors. With the model parameters θ, a user's preference on a certain item can be estimate by $r_{ui} = U_{u\cdot}V_{i\cdot}^T + b_i$, we omit the global bias and user bias as they are reduced in the task of top-N item recommendation for each user.

In general, due to the very large number of item pairs, a common practice is to apply SGD to optimize pairwise preference learning. Uniform sampler has been widely adopted in pairwise preference learning. In practice, it just needs to pick several items from the candidate set for a specific user. However, when the number of items is large, uniform sampler might miss the most of the useful items and select massive ineffective training examples which cannot provide much helpful information to the current prediction model. That is, uniform sampler might slow the convergence of pairwise preference learning, and lead to the model parameters insufficiently learned.

III. Improved Sampling Strategy

A. Alpha-Beta Sampling

To solve this problem mentioned above, we can first revisit the optimization process of pairwise preference learning. Generally, pairwise algorithms learn the rank of user-

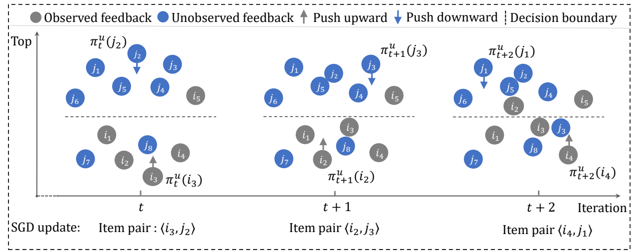

Fig. 1. Illustration to describe the item permutation for user u in different learning stages, where $\pi_u^t(\cdot)$ represents the position of item in iteration t.

s preferred item by minimizing the classification errors of sampled positive-negative item pairs. Before model training, the prediction model parameters are usually initialized with Gaussian distribution or some other manners. Then, sampled training instances will be used to train the target of correctly identifying the positive and negative item of each pair during learning. More specifically, the unobserved items ranking in high positions will be pushed down from the top, and the observed feedbacks ranking in low positions will be optimized to the top. Eventually, a ranked list of the most relevant items will be delivered to the user for each interaction.

Suppose that users preference over items might be insufficiently learned during SGD update. To speed up the convergence of pairwise algorithms, we propose that such examples, for which the model currently predicts very incorrectly, can update the model parameters with a large magnitude based on the above explanation of the process of learning. In other words, such examples including positive item ranking close to the bottom and negative item ranking close to the top can be used as training instances to maximize the utility of the current gradient step. In the following, we name such useful training examples as informative examples. Hence, we propose a novel sampling strategy termed as Alpha-Beta Sampling (ABS) strategy by drawing informative training examples for each gradient step. Furthermore, to formulate the core idea of our proposed strategy, we define a basic assumption for the proposed strategy as follows:

Assumption 2 *For a user u, selecting such training instances, including the negative item j ranking in a high position and the positive item i ranking in a low position, could be a good case to effectively update the model parameters.*

Specifically, an illustration to describe the item permutation based on BPR is showed in Fig 1. This assumption is intuitively correct because the informative examples hurt the ranking performance of the current prediction model more. Even if the higher ranked unobserved item is relevant one, ranking it lower than observed items is still reasonable according to Assumption 1. Thus, more emphasis should be placed on informative examples.

B. Proofs of ABS Strategy

In the following, we provide the corresponding proofs for ABS strategy from two perspectives: gradient update and ranking task.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.

*1) **Gradient Update**:* With sampled training instances S and the tentative objective function $f(u,S)$, pairwise algorithms can be optimized in the update formulation of the SGD, and the update rule is usually represented as follows,

$$\theta = \theta - \gamma(\frac{\partial f(u,S)}{\partial r_{\succ_u}}\frac{\partial r_{\succ_u}}{\partial \theta} + \alpha\theta), \tag{2}$$

where the learning rate γ usually has to be set small enough to ensure that the training step is in the right direction, i.e., the gradient is approximately correct with within a small region around θ. It should be noted that $\alpha\theta$ in Eq. (2) is from a regularization term $\frac{\alpha}{2}||\theta||^2$ used to avoid overfitting. To fit user preference, $f(u,S)$ can be represented in various forms, like Margin Ranking Criterion [8], Fidelity loss [9], Cross Entropy (CE) loss [10] or in other forms. For example, as for pairwise ranking methods based on BPR, the tentative objective function can be written as:

$$\begin{aligned} f(u,S) = &-\ln(1+e^{-r_{\succ_u}}) + \frac{\alpha_u}{2}\|U_{u\cdot}\|^2 + \frac{\alpha_v}{2}\|V_{t\cdot}\|^2 \\ &+ \frac{\beta_v}{2}\sum_{t\in S}\|U_{u\cdot}\|^2, \end{aligned} \tag{3}$$

where $t \in S$ and $S = (i,j)$, $\sigma(x)$ is the sigmoid function, where $\sigma(x) = 1/(1+e^{-x})$. Therefore, the gradients of the parameters w.r.t. the tentative objective function $f(u,S)$ can be reached at:

$$\frac{\partial f(u,S)}{\partial \theta} = \frac{\partial f(u,S)}{\partial r_{\succ_u}}\frac{\partial r_{\succ_u}}{\partial \theta} = (1-\sigma(r_{\succ_u})\frac{\partial \succ_u}{\partial \theta}), \tag{4}$$

note that we omit the bias for clarity. Based above, the influence of sampled user-item set (u,S) can be determined by two parts: the form of the tentative objective function $f(u,S)$, and the quality of sampled item set S. For the tentative objective function, different algorithms usually incorporate different forms of pairwise structures and loss function into the tentative objective function with different goals, which will then result in different values of $\partial f(u,S)/\partial r_{\succ_u}$. Suppose that a specific form of the tentative objective function has been determined, the influence of each gradient step will be mainly determined by the quality of S. Here, we use $\lambda(u,S)$ to represent the gradient $\partial f(u,S)/\partial r_{\succ_u}$. Therefore, we sum up a general preference learning scheme in a concise way as:

$$\textbf{Pairwise Preference Learning}: (g(u,S), \lambda(u,S)). \tag{5}$$

where (i) the first term $g(u,S)$ denotes the utility function of sampled user-item set (u,S). Clearly, $g(u,S)$ can represent how much influence of sampled items S for the current model parameters Θ. (ii) The second term $\lambda(u,S)$ denotes the gradient function, which indicate the influence on how to choose a specific form of pairwise preference structure $\succ_u$ in this tentative objective function $f(\succ_u) = f(u,S)$. With the above learning scheme, the update rule can be equivalently written as follows:

$$\theta = \theta - \gamma(\lambda(u,S)\frac{\partial r_{\succ_u}}{\partial \theta} + \alpha\theta). \tag{6}$$

With the general learning scheme in Eq. (5), we can represent a typical pairwise preference learning algorithm in a concise way.

Two Instantiations. By reviewing the previous works, we summarize that the previous pairwise methods in PROCCF settings can be mainly divided into two groups based on the pairwise preference structure in tentative objective function: (1) standard single-pair pairwise methods [11], [12] , and (2) multiple-pair pairwise methods [5], [13].

a) ABS Strategy for BPR: BPR is a representative single-pair pairwise method. The tentative objective function in BPR can be represented as follows:

$$f(u,S) = \sigma(r_{uij}), \tag{7}$$

where $r_{uij} = r_{ui} - r_{uj}$. Therefore, the scheme of pairwise preference learning based on BPR can be represented as:

$$\textbf{BPR-Scheme:}\ (1-\sigma(r_{uij}), -\frac{1}{1+e^{-r_{uij}}}). \tag{8}$$

With ABS strategy, informative item pair, including positive item i with lower preference and negative item j with higher preference, will have high chance to be sampled as training instances. In this way, the utility function can be closely to 1 and the model parameters can be learned with a large magnitude for this gradient step.

b) ABS Strategy for MPR: Multiple Pairwise Ranking (MPR) adopts to optimize multiple-pair pairwise structure, and it further exploits users' preference between (I) an observed feedback and an unobserved feedback, and (II) two unobserved feedbacks, and (III) two observed feedbacks. The difference value of user's preference between (I) is not less than (II), and (II) is not less than (III). As for MPR, the tentative objective function can be represented as follows:

$$f(u,S) = \sigma(r_{uij} \succ r_{uqq'} \succ r_{upp'}), \tag{9}$$

where item i,p,p' are selected from observed feedback, and item j,q,q' are selected from unobserved feedback. Accordingly, the learning scheme of MPR can be represented as:

$$\begin{aligned} &\textbf{MPR-based Scheme}: \\ &(1-\sigma(r_{uij} \succ r_{uqq'} \succ r_{upp'}), -\frac{1}{1+e^{-r_{uij}\succ r_{uqq}\succ r_{upp'}}}). \end{aligned} \tag{10}$$

By analyzing the learning schemes of MPR, we could find that the utility function will be close to 1 with ABS strategy.

*2) **Ranking Task**:* Previous literatures [11] point out that pairwise methods are to directly optimize the AUC measure. However, such a measure is not a ranking-biased measure and does not reflect well the quality of recommendation lists. Hence, pairwise methods might not match well in top-N recommendation tasks. To tackle this challenge, like the ranking performance optimization in Information Retrieval (IR) tasks, Lambda-based methods [6] have been proposed by incorporating the change of ranking-aware measures, like NDCG, into the pairwise loss. By inheriting the core idea of LambdaRank, we can define a similar lambda-function $h(\lambda(u,S), \zeta_u)$, where ζ_u denotes the current item ranking list

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.

of items for user u. With NDCG as target, $h(\lambda(u, S), \zeta_u)$ can be defined as:

$$h(\lambda(u, S), \zeta_u) \equiv \lambda(u, S) \Delta NDCG_S, \tag{11}$$

where $\Delta NDCG_S$ is the absolute changed NDCG value for the ranked list ζ_u if sampled item in S gets switched together. Then, the gradient update in Eq (6) can be implemented by replacing $\lambda(u, S)$ with $h(\lambda(u, S), \zeta_u)$.

Based above, $\Delta NDCG_S$ can be considered as a learning weight function (i.e., $\lambda(u, S)$) of sampled training instances S for each gradient step. This weight will be raised if the difference of NDCG is larger after swapping the item together, otherwise, it will penalize the training instances by shrinking the learning weight. Suppose there exists an ideal lambda function for each training pair for the current item ranking list, we are able to construct an almost equivalent training model by sampling the training pair with a higher proportion according to the probability (i.e., $\Delta NDCG_S$). It is not hard to observe that informative item pair can be equivalent to such item pair with larger $\Delta NDCG$.

C. Efficient Sampling Algorithm

In this part, we will propose an alternative sampling algorithm to implement ABS strategy in the settings of PROCCF. Without loss of generality, we can implement the ABS strategy by picking items from candidate item set which are already in order [7]. In the settings of PROCCF, Individual user preference can often be evaluated by the preference score function [14]. Considering the ranking scores of items might only have tiny differences, we choose to formalize a predicted rank s_l instead of using the notation of preference score. In this way, the largeness of scores is relative to other examples, but the ranks are absolute values. This allows us to formulate a sampling distribution, e.g., an exponential distribution, based on item ranks such that higher ranked items have a larger chance to be picked.

$$p(l|u) \propto exp(-s_l/\delta), \quad \delta \in R^+, \tag{12}$$

where δ is a hyperparameter controlling the expectation position of sampled item l. Note that we omit the item bias b_i in r_{ul} and it can be described as follows:

$$r_{u,l} = \sum_{f=1}^{k} v_{u,f} v_{l,f} = \sum_{f=1}^{k} |v_{u,f}| sgn(v_{u,f}) v_{l,f}, \tag{13}$$

where $sgn(\cdot)$ denotes the sign function. Then, a transformation $y^*(l)$ of r_{ul} can be define by:

$$y^*(l) = \sum_{f=1}^{k} p(f|u) sgn(v_{u,f}) v_{l,f}, \tag{14}$$

where $p(f|u)$ is the probability function that denotes the importance of the latent dimension f for the user u. Here, we assume that the latent factor is subject to uniform distribution. In order to improve the efficiency of the scoring function, instead of repeating inner product operations, we first randomly sample a latent factor f (i.e., $f \in \{0, 1, ..., k-1\}$) from k

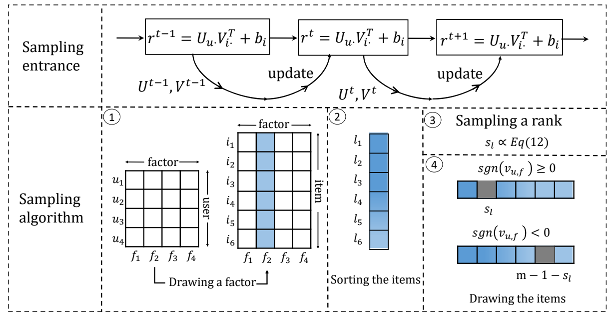

Fig. 2. Illustration of the efficient sampling algorithm.

latent factors. Next, the scoring function under the given item l and latent factor f can be further described as:

$$y(l|u, f) = sgn(v_{u,f}) v_{l,f}. \tag{15}$$

For a given latent vector $v_{u,f}$, the final scoring function $y(l|u, f)$ has the similar effect on the rank of user preference. In other words, the candidate list generated by $y(l|u, f)$ will be approximately equal with those generated by r_{ul}. Then, $y(l|u, f)$ is used to rank all the candidate item.

Based on the above explanation, Furthermore, we illustrate the efficient sampling process and give a specific example in Fig 2. To sum up, the new sampling process can be implemented as follows:

1) A rank s_l is picked from distribution in Eq. (12).
2) Latent factor f is randomly picked from k latent factors.
3) Candidate set is ranked by the scoring function $y(l|u, f)$.
4) The item on rank $s(l|u, f)$ is picked from the candidate set if $sgn(u_{u,f}) \geq 0$, otherwise the item on rank $(m - 1 - s_l)$ is picked.

Time Complexity Analysis. In practice, after a single iteration, the model parameters may change only little and many steps can even be ignored in Eq. (2). Hence, it is unnecessary to update the candidate set for each step while learning. Instead, we choose to rank the candidate set every $|I|log|I|$ iterations. In this way, the running time of this sampling algorithm can be dramatically reduced to $O(k|I|log|I|)$. In summary, the proposed sampling algorithm can well implement ABS strategy by assigning items from candidate item.

Discussion. In our view, ABS strategy can be implemented in various ways, we believe that other instantiations can also achieve similar results. Compared to the previous proposed sampling methods, we propose a new view on pairwise preference learning by considering various pairwise structures. In addition, ABS strategy is more expressive than previous methods since negative sampling methods can be regarded as a special case of ABS strategy without positive item sampling.

IV. Experiments

A. Dataset Description and Evaluation Metrics

Dataset Description. In the experiments, we choose three real-world datasets, i.e., MovieLens100K, MovieLens1M, UserTag to validate the effectiveness of our methods. Static of

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.

TABLE I

RECOMMENDER PERFORMANCE OF BPR-ALPHA, BPR-BETA, BPR-ABS, MPR-ABS, AND BASELINES ON THREE PUBLIC DATASETS.

Dataset	Method	Precision@5	Recall@5	F1@5	1-call@5	MAP	MRR	NDCG@5	AUC
MovieLens100K	MPR	0.3624	0.0865	0.1213	0.7791	0.2435	0.5740	0.3764	0.8945
	BPR	0.3649	0.0909	0.1262	0.7831	0.2475	0.5873	0.3796	0.9046
	AoBPR	0.3715	0.0940	0.1322	0.8036	0.2568	0.6017	0.3874	0.9091
	BPR-Beta	0.3806	0.0957	0.1327	0.8097	0.2569	0.6072	0.3994	0.9095
	BPR-Alpha	0.3732	0.0927	0.1305	0.8025	0.2559	0.6004	0.3914	0.9108
	BPR-ABS	0.3777	0.0968	0.1319	0.8214	0.2551	0.5988	0.3925	0.9112
	MPR-ABS	0.3851	0.0940	0.1309	0.8182	0.2554	0.5933	0.3977	0.9166
MovieLens1M	MPR	0.4147	0.0704	0.1065	0.8315	0.2390	0.6293	0.4286	0.9245
	BPR	0.4122	0.0682	0.1043	0.8265	0.2384	0.6198	0.4249	0.9277
	AoBPR	0.4266	0.0709	0.1082	0.8372	0.2457	0.6282	0.4373	0.9320
	BPR-Beta	0.4289	0.0707	0.1097	0.8416	0.2470	0.6330	0.4374	0.9326
	BPR-Alpha	0.4222	0.0703	0.1075	0.8357	0.2455	0.6292	0.4360	0.9304
	BPR-ABS	0.4308	0.0721	0.1100	0.8456	0.2483	0.6394	0.4429	0.9317
	MPR-ABS	0.4318	0.0727	0.1110	0.8444	0.2493	0.6388	0.4449	0.9332
UserTag	MPR	0.2391	0.0364	0.0565	0.5390	0.1367	0.3866	0.2469	0.7413
	BPR	0.2331	0.0351	0.0552	0.5293	0.1334	0.3825	0.2395	0.7553
	AoBPR	0.2456	0.0373	0.0579	0.5296	0.1400	0.3852	0.2502	0.7573
	BPR-Beta	0.2520	0.0386	0.0602	0.5533	0.1428	0.3907	0.2528	0.7545
	BPR-Alpha	0.2532	0.0373	0.0591	0.5373	0.1413	0.4011	0.2590	0.7597
	BPR-ABS	0.2587	0.0397	0.0626	0.5533	0.1445	0.3946	0.2611	0.7594
	MPR-ABS	0.2492	0.0387	0.0599	0.5495	0.1393	0.4020	0.2569	0.7427

three datasets are summarized in Table II. For MovieLens100K and MovieLens1M, we only keep the ratings larger than three points as the positive feedback (to simulate the one-class feedback). For all three datasets, the observed user-item pairs are randomly split into two parts, i.e., one as training data, and the other as test data. Meanwhile, we randomly sample one user-item pair for each user from the training data to construct a validation set. We repeat the above procedure 10 times. The final experimental results are averaged over every evaluation metric on these 10 copies of test data.

TABLE II

DESCRIPTION OF THE EXPERIMENTAL DATASETS.

Datasets	MovieLens100K	MovieLens1M	UserTag
#Ratings	100,000	1,000,209	246,436
#Training instances	27,688	287,641	123,218
#Test instances	27,687	287,641	123,218
#Users	943	6,040	3,000
#Items	1,682	3,952	2,000

Evaluation Metrics. We adopt commonly used top-N evaluation metrics, including Top-5 results of *Precision*, *Recall*, *F1* and *1-call*. We also adopt ranking-aware evaluation metrics, including *MAP*, *MRR* and *NDCG@5*. In addition, *AUC* is included as a reference which is used in BPR.

B. Baselines and Parameter Settings

Baselines. To evaluate the performance of ABS strategy, we borrow some baselines from various perspectives. Specifically, we compare the ABS strategy with uniform sampler applied in two representative methods, i.e., BPR [11] and MPR [5]. Also, a state-of-the-art sampling strategy (i.e., AoBPR) is also included. We also conduct a supplementation experiment regarded with the variants of our strategies:

- **BPR-Alpha**: Positive sampling draws the positive item as in ABS strategy and the other items as in uniform sampling strategy for each gradient.
- **BPR-Beta**: Negative sampling draws the negative item as in ABS strategy and the other item as in uniform sampling strategy for each gradient step.
- **BPR-ABS**: Both positive and negative item sampling according to ABS strategy.
- **MPR-ABS**: Positive and negative item sampling by adopting ABS strategy.

Parameters Settings. We search the regularization terms as $\varphi_u, \varphi_v, \phi_v \in \{0.001, 0.01, 0.1\}$, and δ of the geometric distribution is searched around $\{500, 5000, 50000\}$, and the iteration step T is chosen in the range of $\{1000, 10000, 100000\}$. We select the best parameter $\varphi_u, \varphi_v, \phi_v$ and the best iteration step T for all algorithms according to the *Recall* performance on the validation data. For the learning rate of γ, we fix it as 0.01. The number of latent dimensions in matrix factorization is fixed as $k = 20$.

C. Results and Analysis

Ranking Performance Analysis. The overall ranking performance of all methods on three datasets are shown in Table I. From the overall views, our strategy has achieved the best performances. To some extent, it verifies that pairwise approaches by adopting our strategy is approximate to construct an e-quivalent training model for optimizing rankingaware measure. Also, we notice that several variants of ABS strategy applied in BPR also achieve comparative better performance than uniform sampler. It means that utilizing the rank information of positive and negative item sampling indeed can lead to an improvement of pairwise ranking. In addition, we also find some interesting evidences. Negative item sampling can lead to more improvements than positive item sampling. The main reason may include that the size of the observed item set is different from the unobserved ones.

Convergence Analysis. Due to the limited space, we only list the learning curves of different sampling strategies of BPR

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.

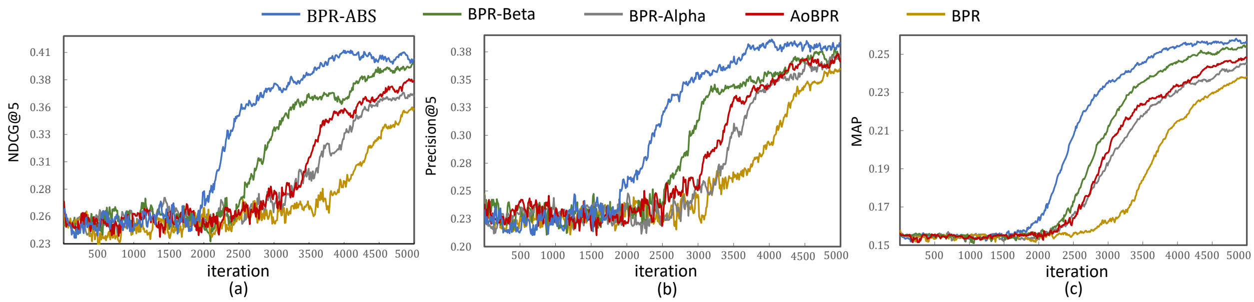

Fig. 3. Learning curves of BPR with different sampling strategies on MovieLens100K.

methods on MovieLens100K. Fig 3 (a), (b), (c) show how the learning curves of different methods varies given different e-valuation metrics. We can find that our proposed ABS strategy achieve the best learning effectiveness compared to baseline strategies, which indicates that ABS can be more effective and powerful for the sampling task of BPR by drawing positive and negative items in a fine-grained way. Meanwhile, the variant of ABS applied in BPR, i.e., BPR-Beta also have better performance than other baselines. Specifically, first, all variant of ABS strategy performs better than uniform sampler, which indicates non-uniform sampling strategies can achieve better results than the uniform sampler. Second, individual positive item sampling does not perform as well as negative item sampling. We think the main reason is that the observed item is not as large as the unobserved item space, and the improvement is limited to a certain extent. Third, it also can be seen that all algorithms almost converge after a number of iterations, then fluctuate in a tiny range around.

V. Conclusions

To solve the ineffectiveness problem of traditional sampling methods, we first introduce Alpha-Beta Sampling strategy by drawing such examples for which the current model predicts incorrectly. Different from previous strategies, we proved that our proposed strategy is more general and applicable to various pairwise structures. Further, it can be proved that our proposed strategy is approximately to optimize ranking-aware measures from top-N recommendation perspective. After that, we propose an alternative instantiation to implement ABS strategy by leveraging the latent vectors. In our view, the exact form of the sampling algorithm is not crucial, we hold that more specific sampling algorithms can also achieve similar results by adopting the idea of ABS strategy. Finally, extensive experimental results on real-world social network datasets demonstrated the effectiveness of our strategy. We hope this study could lead to more future work.

VI. Acknowledgements

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 61672483, 61922073, U1605251). Qi Liu gratefully acknowledges the support of the Young Elite Scientist Sponsorship Program of CAST and the Youth Innovation Promotion Association of CAS (No. 2014299).

References

[1] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, "Personalized travel package recommendation," in *2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 407–416.

[2] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, "Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1734–1743.

[3] Z. Ye, L. Zhang, K. Xiao, W. Zhou, Y. Ge, and Y. Deng, "Multi-user mobile sequential recommendation: An efficient parallel computing paradigm," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2624–2633.

[4] Q. Liu, H.-K. Zhao, L. Wu, Z. Li, and E.-H. Chen, "Illuminating recommendation by understanding the explicit item relations," *Journal of Computer Science and Technology*, vol. 33, no. 4, pp. 739–755, 2018.

[5] R. Yu, Y. Zhang, Y. Ye, L. Wu, C. Wang, Q. Liu, and E. Chen, "Multiple pairwise ranking with implicit feedback," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 1727–1730.

[6] W. Zhang, T. Chen, J. Wang, and Y. Yu, "Optimizing top-n collaborative filtering via dynamic negative item sampling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 785–788.

[7] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 273–282.

[8] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng, "Collaborative competitive filtering: learning recommender using context of user choice," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 295–304.

[9] M.-F. Tsai, T.-Y. Liu, T. Qin, H.-H. Chen, and W.-Y. Ma, "Frank: a ranking method with fidelity loss," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 383–390.

[10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, 2005, pp. 89–96.

[11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[12] W. Pan and L. Chen, "Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering." in *IJCAI*, vol. 13, 2013, pp. 2691–2697.

[13] J. Ding, G. Yu, X. He, Y. Li, and D. Jin, "Sampler design for bayesian personalized ranking by leveraging view data," *arXiv preprint arXiv:1809.08162*, 2018.

[14] Q. Liu, E. Chen, H. Xiong, C. H. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 218–233, 2011.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 16,2020 at 13:17:50 UTC from IEEE Xplore. Restrictions apply.