Adaptive Pairwise Preference Learning for Collaborative Recommendation with Implicit Feedbacks

Hao Zhong†, Weike Pan‡, Congfu Xu‡*, Zhi Yin§ and Zhong Ming¶
†Institute of Artificial Intelligence, College of Computer Science, Zhejiang University
‡College of Computer Science and Software Engineering, Shenzhen University
¶College of Science, Ningbo University of Technology

{haozhong,xucongfu}@zju.edu.cn, {panweike,mingz}@szu.edu.cn, yz@nbut.edu.cn
* Corresponding author

ABSTRACT

Learning users’ preferences is critical to enable personalized recommendation services in various online applications such as e-commerce, entertainment and many others. In this paper, we study on how to learn users’ preferences from abundant online activities, e.g., browsing and examination, which are usually called implicit feedbacks since they cannot be interpreted as users’ likes or dislikes on the corresponding products directly. Pairwise preference learning algorithms are the state-of-the-art methods for this important problem, but they have two major limitations of low accuracy and low efficiency caused by noise in observed feedbacks and non-optimal learning steps in update rules. As a response, we propose a novel adaptive pairwise preference learning algorithm, which addresses the above two limitations in a single algorithm with a concise and general learning scheme. Specifically, in the proposed learning scheme, we design an adaptive utility function and an adaptive learning step for the aforementioned two problems, respectively. Empirical studies show that our algorithm achieves significantly better results than the state-of-the-art method on two real-world data sets.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

Keywords

Collaborative Recommendation; Implicit Feedbacks; Pairwise Preference Learning

1. INTRODUCTION

Recommendation and personalization technology has an extremely wide spectrum of online applications, including e-commerce, entertainment, professional networks, mobile advertisement, etc. Automatically mining and learning users’ preferences from their online activities such as browsing and examination records is critical to provide qualified personalized services. Such activities are usually called users’ implicit feedbacks, which is often of low efficiency due to the resulted non-optimal learning steps. Some works have realized the above two problems and relax the pairwise relationships via introducing a new preference score on a set of items [2] instead of on a single item [5], which introduces a more general loss function. Some other works design some advanced sampling strategies for the second issue such as [4].

In this paper, we aim to address the aforementioned two problems in one single algorithm. Specifically, we design a concise and general learning scheme, which is able
to absorb different loss functions and sampling strategies as special cases. Furthermore, we design an adaptive utility function and learning step in a pairwise preference learning algorithm, which is thus called APPLE (adaptive pairwise preference learning).

2. OUR SOLUTION: ADAPTIVE PAIRWISE PREFERENCE LEARNING

2.1 Problem Definition

We use \( \mathcal{R} = \{(u, i)\} \) to denote a set of implicit feedbacks or activities from \( n \) users and \( m \) items. Each (user, item) pair \((u, i)\) means that user \( u \) has browsed or examined item \( i \), which is usually called an implicit feedback of user \( u \) on item \( i \) due to the uncertainty of the user’s true preference. Our goal is then to exploit the data \( \mathcal{R} \) in order to generate a personalized ranked list of items from \( \{j | (u, j) \notin \mathcal{R}\} \) for each user \( u \).

2.2 A General Learning Scheme

A pairwise preference learning algorithm usually minimizes a tentative objective function \( f(u, i, j) \) for a randomly sampled triple \((u, i, j)\). A triple \((u, i, j)\) means that the relationship between user \( u \) and item \( i \) is observed while the relationship between user \( u \) and item \( j \) is not observed. In order to encourage pairwise competition, the tentative objective function is usually defined on a pairwise preference difference, i.e., \( f(u, i, j) = f(\hat{r}_{uij}) \), where \( \hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj} \) is the difference between user \( u \)’s preferences on item \( i \) and item \( j \). A user \( u \)’s preference vector on an item \( i \), i.e., \( \hat{r}_{ui} \), is typically modeled by a set of parameters denoted by \( \theta \), which include user \( u \)’s latent feature vector \( \hat{U}_u \in \mathbb{R}^{1 \times d} \), item \( i \)’s latent feature vector \( \hat{V}_i \in \mathbb{R}^{1 \times d} \) and item \( i \)’s bias \( b_i \in \mathbb{R} \). With the model parameter \( \theta \), we can estimate a user’s preference on a certain item via \( \hat{r}_{ui} = \hat{U}_u \hat{V}_i^T + b_i \).

With a sampled triple \((u, i, j)\) and a tentative objective function \( f(\hat{r}_{uij}) \), the model parameter \( \theta \) can then be learned or updated accordingly. The update rule is usually represented as follows,

\[
\theta = \theta - \gamma \left( \frac{\partial f(S_{uij})}{\partial \hat{r}_{uij}} \frac{\partial r_{uij}}{\partial \theta} + \alpha \theta \right). \tag{1}
\]

where \( f(\hat{r}_{uij}) \) can be \(-\ln 1/(1 + e^{-\hat{r}_{uij}}) \) [5], \( \max(0, 1 - \hat{r}_{uij}) \) [6] or in other forms, in order to encourage different types of pairwise competitions between an observed pair \((u, i)\) and an unobserved pair \((u, j)\). Note that \( \alpha \theta \) in Eq.(1) is from a regularization term \( \sum \frac{1}{2} \| \theta \|_2^2 \) used to avoid overfitting.

There are two fundamental questions associated with the update rule in Eq.(1), namely (i) how to choose a specific form of the tentative objective function \( f(\hat{r}_{uij}) \), and (ii) how to sample a triple \((u, i, j)\). For the first question, different works usually incorporate different loss functions into \( f(\hat{r}_{uij}) \) with different goals, which will then result in different values of \( \frac{\partial f(S_{uij})}{\partial \hat{r}_{uij}} \). For the second question, most previous works sample a triple in a uniformly random manner [5, 6].

Mathematically, the above two questions can be represented by a concise and general learning scheme,

**Learning Scheme:** \( (\rho(u, i, j), \tau(u, i, j)) := (\rho, \tau) \) \tag{2}

where (i) the first term \( \rho(u, i, j) \) denotes the utility of a randomly sampled triple \((u, i, j)\), which answers the question of how to sample a triple, and (ii) the second term \( \tau(u, i, j) = \rho(u, i, j) \tau(u, i, j) \) answers the question of how to choose a specific form of the tentative objective function.

The update rule in Eq.(1) can then be equivalently written as follows,

\[
\theta = \theta - \gamma \left( \tau(u, i, j) \frac{\partial r_{uij}}{\partial \theta} + \alpha \theta \right). \tag{3}
\]

where \( \frac{\partial r_{uij}}{\partial \theta} \) is the gradient, which answers the question of how to choose a specific form of the tentative objective function.

The learning scheme \( (\rho(u, i, j), \tau(u, i, j)) \) in Eq.(2) for pairwise preference learning can also be described by an algorithm, which is shown in Figure 1, in particular of lines 5-7. In Figure 1, we can see that the chance of sampling a triple \((u, i, j)\) is \( \rho(u, i, j)/|\mathcal{T}| \), where \(|\mathcal{T}|\) is the number of triples in \( \mathcal{T} \) and \( \rho(u, i, j) \) is the utility of the randomly sampled triple.

Figure 1: The algorithm of adaptive pairwise preference learning (APPLE).

**Input:** Triples \( \mathcal{T} = \{(u, i, j)\}_{1 \leq u \leq n, 1 \leq i, j \leq m} \), and learning scheme \( (\rho(u, i, j), \tau(u, i, j)) \).

**Output:** Model \( \Theta = \{U_u, V_i, b_i\}_{1 \leq u \leq n, 1 \leq i \leq m} \).

1: for \( t = 1, \ldots, T \) do
2: \hspace{1em} \text{repeat}
3: \hspace{2em} Randomly sample a triple \((u, i, j)\) from \( \mathcal{T} \).
4: \hspace{2em} Generate a random variable \( \rho_{rand} \in [0, 1] \).
5: \hspace{2em} Calculate the utility \( \rho(u, i, j) \).
6: \hspace{2em} until \( \rho_{rand} \leq \rho(u, i, j) \)
7: \hspace{2em} Update model via Eq.(3) with \( \tau(u, i, j) \).
8: \hspace{2em} if \( S-II \& mod (t, K) = 0 \) then
9: \hspace{3em} Update \( \tau_2 \) via Eq.(7).

2.3 Two Specific Learning Schemes

It is well known [4] that a large preference difference \( \hat{r}_{uij} \) means that the pairwise competition between \((u, i)\) and \((u, j)\) of a typical triple \((u, i, j)\) has been well encouraged, and thus may not be helpful to use this \( \hat{r}_{uij} \) in the update rule in Eq.(1). This observation motivates us to sample triples with small preference differences. We thus propose a preliminary learning scheme with an adaptive utility function without changing the expectation of the learning step.

\[
\text{S-BPR:} \quad \left(1, -\frac{e^{-\hat{r}_{uij}}}{1 + e^{-\hat{r}_{uij}}} \right) := (\rho_{BPR}, \tau_{BPR}), \tag{4}
\]

from which we can see that our learning scheme in Eq.(2) is quite powerful and is able to absorb other pairwise preference learning algorithms as special cases.

Based on the general learning scheme in Eq.(2), we propose one preliminary learning scheme and two specific learning schemes so as to learn users’ true preferences in a more effective and efficient way.

**Learning Scheme (S-BPR):** \( (\rho_{BPR}, \tau_{BPR}) := (\rho_0, \tau_0) \). \tag{5}

It is easy to show that a smaller \( \hat{r}_{uij} \) will result in a larger utility, and the expectation of the learning step \( |\mathcal{T}| \) for \((u, i, j)\)
of S-0 and S-BPR are the same, \( \frac{1}{|\tau|} \times \rho_{BPR} \times |\tau_{BPR}| = \frac{1}{|\tau|} \times \rho_0 \times |\tau_0| \). The advantages of the learning scheme S-0 as compared with S-BPR are, (i) a triple \((u, i, j)\) with larger \(\tau_{uij}\) will have a lower chance to be sampled, and (ii) the learning step \(|\tau_0|\) is larger than \(|\tau_{BPR}|\), which is assumed to be helpful for the learning efficiency.

In the following sections, we will describe two specific learning schemes based on this preliminary learning scheme.

2.3.1 Scheme I

We assume that a triple \((u, i, j)\) with a very small \(\tau_{uij}\) often denotes a high chance that user \(u\) dislikes item \(i\) or user \(u\) likes item \(j\), and then we may not encourage the pairwise competition between \((u, i)\) and \((u, j)\) any more. Hence, such a triple \((u, i, j)\) is considered noise for pairwise preference learning. As a response, we design a new utility function, \(\rho(u, i, j) = \frac{1}{1 + e^{-\frac{\tau_{uij}}{2}}} \times \frac{e^{-r_{uij}}}{1 + e^{-r_{uij}}} = \frac{e^{-r_{uij}}}{(1 + e^{-r_{uij}})^2}\), in order to reduce the chance that a triple \((u, i, j)\) with lower \(\tau_{uij}\) will be sampled. This new utility function reaches the peak value 0.25 when \(\tau_{uij} = 0\), and becomes smaller as \(\tau_{uij}\) increases or decreases. In order to constrain the value range to \([0, 1]\), we obtain our first learning scheme,

\[
\text{S-I: } \left( \frac{4e^{-r_{uij}}}{(1 + e^{-r_{uij}})^2}, -1 \right) := (\rho_1, \tau_1). \tag{6}
\]

We can see that the difference between S-I and S-0 is the utility \(\rho(u, i, j)\), i.e., \(\rho_1\) and \(\rho_0\), where \(\rho_1\) is designed to reduce the impact of noisy triples (i.e., triples with small preference difference \(\tau_{uij}\) when the learning process has conducted for some time).

2.3.2 Scheme II

In order to improve the learning efficiency, we may set the learning step \(|\tau|\) to be a larger value in the beginning, since most \(\tau_{uij}\) are very small. While in the middle or in the end of the learning process, we shall decrease the learning step \(|\tau|\) so as to reach the optimal solution in a smooth manner and thus to achieve high recommendation accuracy. We then propose a more sophisticated learning scheme accordingly in order to benefit both learning efficiency with large \(|\tau|\) and recommendation accuracy with small \(|\tau|\).

We first show the distribution of preference difference \(\tilde{r}_{uij}\) of Movielens1M data (see more information in the Section of Experimental Results) in different learning stages in Figure 2. We can see that the whole distribution will move from the origin to the right, which means that the difference \(\tilde{r}_{uij}\) becomes larger in the learning process as expected by the competition encouragement. Based on this observation, we propose to use the average preference difference to update the value of \(|\tau|\) in a certain monotonic decreasing function. Due to the complexity of \(\rho_1\) and the fact that \(\tilde{r}_{uij} \geq 0\) in most cases, we use \(\rho_1' = \frac{e^{-\gamma(\tilde{r}_{uij})}}{1 + e^{-\gamma(\tilde{r}_{uij})} - \gamma} \) to approximate \(\rho_1\) when \(\tilde{r}_{uij} \in [0, \infty)\) in Eq. (6), and obtain an estimation \(x = 2, y = 1.5\) via minimizing the KL-divergence between \(\rho_1\) and \(\rho_1\) with \(\tilde{r}_{uij} \geq 0\). We then have \(\tilde{r}_{uij} = \frac{\ln((\gamma + 1)/\gamma)}{1 + e^{-\gamma(\tilde{r}_{uij})}} + 2 \approx \frac{\ln(N/1.5)}{1 + e^{-\gamma(\tilde{r}_{uij})}} + 2\).

\[
\text{S-II: } \left( \frac{4e^{-\tilde{r}_{uij}}}{(1 + e^{-\tilde{r}_{uij}})^2}, -\ln(g+1-\tilde{r}_{uij})-1 \right) := (\rho_2, \tau_2), \tag{7}
\]

where \(g\) is set as the maximal value of \(\tilde{r}_{uij}\) obtained when the learning process converges so as to ensure that \(\tau_2\) is larger than 1 in the learning process and is roughly equal to 1 in the end. We can see that the major difference between S-II and S-I is the gradient \(\tau\), which is not static and fixed as \(-1\) in Eq. (6), but is dynamic w.r.t. the preference difference. The new gradient \(\tau\) aims to achieve better learning efficiency and recommendation accuracy, which is also supported by our empirical studies.

3. EXPERIMENTAL RESULTS

3.1 Data Sets

In our empirical studies, we use two data sets, including Movielens1M\(^1\) and Douban\(^2\).

**Movielens1M** Movielens1M contains about 1 million triples in the form of (user, movie, rating) with \(n = 6,040\) users and \(m = 3,952\) movies. In our experiments, we randomly take about 50% triples as training data, about 10% triples as validation data, and the remaining about 40% triples as test data. For training data, we keep all triples and take the corresponding (user, movie) pairs as implicit feedbacks. For validation data and test data, we only keep triples with ratings equal to 4 or 5 and take the corresponding (user, movie) pairs as implicit feedbacks [3].

**Douban** We crawled a real implicit data of users’ feedbacks on books from Douban.com in December 2013, which is one of the largest Chinese online social media websites. The Douban data contains about 3 million (user, book) reading records from 10,000 users and 10,000 books. Similarly, we randomly take about 50% records as training data, about 10% records as validation data and the remaining about 40% records as test data.

We conduct the above “50%, 10%, 40%” splitting procedure of each data for 5 times and thus get 5 copies of training data, validation data and test data.

3.2 Evaluation Metric

We adopt a commonly used top-\(k\) evaluation metric for implicit feedbacks [3], i.e., precision (Pre@k). We use \(k = 5\).

\(^1\)http://grouplens.org/datasets/movielens/
\(^2\)http://www.douban.com/
3.3 Baselines and Parameter Settings

In our experiments, we study our proposed algorithm APPLE with two specific learning schemes in comparison with the state-of-the-art algorithm BPR (Bayesian personalized ranking) [5]. We implement both learning schemes in Eq.(6) and Eq.(7) and that of BPR in Eq.(4) in the same algorithmic framework in Figure 1 for fair comparison.

For all experiments, we use the validation data and Pre@5 to tune the hyperparameters. Specifically, we search the regularization parameter $\alpha$ in the range of $[0.001, 0.5]$. For the parameter $g$ of our learning scheme S-II in Eq.(7), we have tried $g \in \{1, 2, 3, 4\}$ to find an approximation of the maximal value of $\tau_{uij}$ in Eq.(7). For the learning rate $\gamma$ in Eq.(3), we fix it as 0.01 [3]. The parameter $K$ is set as $10^5$, and the iteration number $T$ is searched around $10^8$ for sufficient convergence. The number of latent features for users and items is fixed as $d = 10$ for MovieLens1M and $d = 20$ for Douban.

3.4 Summary of Experimental Results

The recommendation performance of our two learning schemes and S-BPR are shown in Table 1, from which we can see that the overall recommendation performance ordering is S-II > S-S-P. We also conduct significance test and find that S-I and S-II are significantly better than S-BPR on both data sets. The results in Table 1 clearly demonstrates the advantages of our proposed learning schemes in our algorithm APPLE, in particular of the learning scheme S-II with sophisticated utility $\rho(u, i, j)$ and dynamic learning step $|\tau(u, i, j)|$.

The learning efficiency of our two learning schemes and S-BPR are shown in Figure 3, from which we have a similar observation, i.e., the overall convergence performance ordering is S-II > S-I > S-BPR. It is interesting to see that this ordering is consistent with that of the learning step of the learning schemes, i.e., $|\tau_2| > |\tau_1| > |\tau_{BPR}|$, which means that increasing the learning step can indeed improve the convergence performance.

Table 1: Recommendation accuracy of our proposed algorithm APPLE with learning schemes S-I in Eq.(6) and S-II in Eq.(7), and the seminal algorithm BPR [5] with S-BPR in Eq.(4) on MovieLens1M and Douban data sets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MovieLens1M</th>
<th>Douban</th>
</tr>
</thead>
<tbody>
<tr>
<td>PopRank</td>
<td>0.2322±0.0037</td>
<td>0.2845±0.0012</td>
</tr>
<tr>
<td>BPR</td>
<td>0.3403±0.0030</td>
<td>0.3821±0.0032</td>
</tr>
<tr>
<td>APPLE(I)</td>
<td>0.3483±0.0026</td>
<td>0.3976±0.0023</td>
</tr>
<tr>
<td>APPLE(II)</td>
<td>0.3497±0.0044</td>
<td>0.3983±0.0013</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel algorithm called adaptive pairwise preference learning (APPLE) for collaborative recommendation with implicit feedbacks. Our proposed algorithm improves the state-of-the-art pairwise preference learning algorithm, i.e., BPR [5], via a concise and general learning scheme with an adaptive utility function and an adaptive learning step. Empirical studies show that our algorithm performs significantly better than the BPR algorithm regarding both the recommendation accuracy and learning efficiency.

For future work, we are mainly interested in generalizing our learning scheme in APPLE to include heterogeneous user feedbacks and social context information.

5. ACKNOWLEDGMENT

We thank the support of National Natural Science Foundation of China (NSFC) No. 61272303, National Basic Research Program of China (973 Plan) No. 2010CB327903, Natural Science Foundation of SZU No. 201436, NSFC No. 61170077, NSF GD No. 1035180601000000, GD S&T No. 2012B091100198, S&T of SZ No. JCYJ20130326110956468 and No. JCYJ20120613102309248. Natural Science Foundation of Ningbo No. 2012A610029 and Department of Education of Zhejiang Province(Y201120179).

6. REFERENCES