저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer
Bayesian Personalized Ranking
with Count Data

by

Dongwoo Kim

A Dissertation
submitted in fulfillment of the requirement
for the degree of
Master of Science
in
Statistics

The Department of Statistics
College of Natural Sciences
Seoul National University
August, 2017
Abstract

Dongwoo Kim
The Department of Statistics
The Graduate School
Seoul National University

Bayesian personalized ranking (BPR) is one of the state-of-the-art models for implicit feedback. Unfortunately, BPR has a limitation that it considers only the binary form of implicit feedback. In this paper, in order to overcome the limitation, we suggest an adapted version of BPR regarding the numeric value of implicit feedback like count data. Furthermore, we implement our model and original BPR in R and compare the results. This model may be useful to reflect implicit feedback more intensively than BPR.

Keywords: Recommendation system, Implicit feedback, Count data, Matrix factorization, Bayesian personalized ranking (BPR).

Student Number: 2015-22579
Contents

1 Introduction ........................................... 1

2 Preliminaries ......................................... 3
   2.1 Implicit Feedback .................................. 3
   2.2 Matrix Factorization (MF) ................. 4
   2.3 Matrix Factorization for Implicit Feedback ... 8
   2.4 Bayesian Personalized Ranking (BPR) .... 9

3 Bayesian Personalized Ranking with Count Data ... 14
   3.1 Notation ......................................... 14
   3.2 Model ............................................ 15
   3.3 Meaning of $c_{ui}$ .............................. 16
   3.4 SGD Algorithm ................................. 18
   3.5 Implementation ................................. 19

4 Discussion ........................................... 24

Reference .............................................. 25

Abstract in Korean .................................. 27
Chapter 1

Introduction

 Recommending items automatically is important and useful task in many e-commerce. Especially, personalized recommendation has been the main topic of recommendation systems. For example, Amazon, one of the biggest online shopping companies, recommends items for each customer. Netflix, the company providing movies in online, recommends movies for each customer. Both companies have improved sales by personalized recommendation. The goal of personalized recommendation system is to predict the user’s preference about item considering the user’s historical data like rating, trading, clicks, viewing history, etc.

 In recommendation system, there are two main types of data: explicit and implicit feedback. Explicit feedback is user’s preference itself like 1-5 scale rating. On the other hand, implicit feedback is user behavior which reflect indirect preference like view, click, buying history. There are many methods to predict user’s preference from explicit feedback, such as item-based collaborative filtering in Sarwar et al. (2001) matrix factorization in Koren, Bell, and Volinsky (2009). In many cases, however, implicit feedback is easier to gather
and much more than explicit feedback. In extreme case, there may be no ex-

plicit feedback. Therefore, it is important to recommend items from implicit
feedback.

Of course, many recommendation methods for implicit feedback have been
studied. Hu, Koren, and Volinsky (2008) suggested the model of matrix factor-
ization regarding implicit feedback as the confidence level. Rendle et al. (2009)
suggested Bayesian personalized ranking (BPR) which is the probabilistic
model about user’s preference of item pairs. However, the original BPR han-
dles only the binary form of implicit feedback, i.e. BPR only considers whether
an user provided feedback about an item or not. Therefore, BPR exploits only
a part of implicit feedback when the data is given as the non-binary form of
implicit feedback, such as the count data which is the typical form of implicit
feedback.

In this paper, we suggest a variation of BPR exploiting the numeric value
of implicit feedback. We expect that our model may be useful to handle the
whole of implicit feedback. We also implemented our model and original BPR
in R and compared the evaluation results.

The remainder of this paper is organized as follows. Chapter 2 reviews
implicit feedback, matrix factorization for explicit feedback, matrix factoriza-
tion for implicit feedback and BPR. Chapter 3 describes our model which is
a variation of BPR for non-binary implicit feedback and the results of imple-
mentation. Chapter 4 provides summary and discussion.
Chapter 2

Preliminaries

In this chapter, we review preliminaries: implicit feedback, matrix factorization, matrix factorization for implicit feedback and Bayesian personalized ranking.

2.1 Implicit Feedback

In this section, we explain the meaning of implicit feedback and some characteristics of it. In recommendation system, there are two types of main input: explicit feedback and implicit feedback. *Explicit feedback* is user’s preferences of items itself. For example, users in Netflix present their preference of movie as the 1-5 score. In Youtube and Facebook, users present their preferences of contents as ‘like/dislike’. On the other hand, *implicit feedback* is the user behavior which reflects their interest indirectly. For example, purchase history is implicit feedback. Suppose there is a user who bought an item. The user may have preferred to buy an item, but it is not sure only by the purchase history. There are many other types of implicit feedback, such as click, view, etc.
There are many characteristics of implicit feedback, but we describe only two important things in this paper.

- Implicit feedback consists of only non-negative feedback. Note that explicit feedback can be positive or negative feedback. In the 1-5 score, a score of 5 may mean 'good', and a score of 1 may mean 'bad'. In like/dislike feedback, it is more obvious. However, the numerical value of implicit feedback is usually count or time of user’s positive behavior. Therefore the low number of frequency doesn’t mean negative feedback. Notice that recommendation system can be regarded as a classification problem. And it is a big problem in classification that there is no negative observation at all.

- The numerical value of implicit feedback indicates the level of confidence about the feedback. As we referred, the numerical value of implicit feedback is usually frequency of user behavior. For example, let’s suppose that a user purchased a food once. It is not sure whether the user is satisfied. However, if the user has purchased the same food for 10 times, it is quite sure that the user loves it. That is, the value of implicit feedback gives us confidence level about the feedback.


2.2 Matrix Factorization (MF)

In this section, we review matrix factorization (MF) for explicit feedback in Koren, Bell, and Volinsky (2009). MF is one of the very popular models of recommendation system since when it won the Netflix prize. When explicit
feedback is given in the form of rating, the main goal of the recommendation system is to predict all user’s unobserved ratings. Then, it is possible to recommend items by the order of predicted ratings. The problem is that there are only a few observed ratings than unobserved ones since customers actually rate a few items compared to the number of all items, and this situation usually called as sparsity. The latent factor model often appears to reduce the dimension of variables or to handle the sparse data. And MF is one of the latent factor models in the recommendation system. MF assumes that there are latent factors about users and items, then predicts ratings by these factors.

In formal, let \( U \) be the set of all users and \( I \) be the set of all items. Define \( r_{ui} \) as the rating which user \( u \) rated for item \( i \), and \( R = (r_{ui})_{(u,i) \in U \times I} \) as the \(|U| \times |I|\) rating matrix for all users and items. Define the set of user-item pairs whose related rating is observed as \( S := \{(u,i) \in U \times I : r_{ui} \text{ is observed}\} \). Note that \( R \) is very sparse matrix, since only \( r_{ui} \) for \((u,i) \in S\) are observed. Our goal is to predict \( r_{ui} \) for \((u,i) \in S^c\).

MF with \( k\)-dim factors assumes that \( r_{ui} \) is obtained by

\[
r_{ui} = p^T_u q_i
\]  

(2.1)

or

\[
r_{ui} = \mu + b_u + b_i + p^T_u q_i,
\]  

(2.2)

where \( p_u, q_i \in \mathbb{R}^k \) are latent factors of user \( u \) and item \( i \), and \( \mu \) is overall average of ratings, \( b_u, b_i \) are the main effects of user \( u \) and item \( i \). \( b_u, b_i \) are usually called biases or baselines in the field of recommendation system.

To understand the intuition of MF, now we focus on the equation (2.1). MF predicts ratings \( r_{ui} \) as the inner product of user factor \( p_u \) and item factor \( q_i \). Figure 2.1 illustrates how MF predicts the ratings when all factors are given.
Figure 2.1: An illustration of how MF predicts ratings. Rating matrix $R$ is decomposed to two low rank matrices $P$ and $Q$.

Suppose there are users \{Alice, Bob, Charlie, \ldots\}, movies \{Titanic, Star Wars, Love Actually, \ldots\} and three factors \{comedy, SF, romance\}. $P$ is the user factor matrix whose rows are $p_u^T$s. Similarly, $Q$ is the item factor matrix whose rows are $q_i^T$s. The $j$-th component of $q_i$ indicates the extent of item $i$’s $j$-th factor, e.g., *Love Actually*’s SF factor is 0, so there is no SF factor in *Love Actually*. Similarly, the $j$-th component of $p_u$ indicates the extent of user $u$’s preference about factor $j$, e.g., Alice’s comedy factor is 1 and others are 0, so we can say that user Alice likes comedy. Note that what we explained about factors is just for understanding. In fact, it is difficult to interpret what factors mean.

The loss function of MF with $\ell_2$-regularization is defined as

$$L(P, Q) = \frac{1}{2} \sum_{(u,i) \in S} (r_{ui} - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2)$$

(2.3)
where $P$, $Q$ are the factor matrix whose rows are $p_u^T$s and $q_i^T$s respectively. $\lambda$ is a regularization parameter and $\| \cdot \|^2$ means the sum of squares of all elements in the argument. The parameters are obtained by minimizing the loss $L(P, Q)$, i.e.

$$
(\hat{P}, \hat{Q}) = \arg\min_{P, Q} \frac{1}{2} \sum_{(u,i) \in S} (r_{ui} - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2).
$$

(2.4)

The stochastic gradient descent (SGD) or alternating least squares (ALS) method is used for optimizing problem in the equation (2.4).

Since SGD is used for optimizing method in BPR and our model, we briefly explain the algorithm of SGD in here. SGD is basically gradient descent (GD) algorithm. For each iteration in GD, parameters are updated as

$$
\theta^{\text{new}} = \theta^{\text{old}} - \eta \nabla L(\theta^{\text{old}}).
$$

Suppose the loss function is the sum of losses for each sample, such as the loss in (2.3). To apply GD, it is needed to calculate the gradient of the loss function. When the number of samples is too large or the shape of gradient is complex, it is expensive to calculate the gradient of the loss for all samples. SGD solves this problem by using the gradient for one sample. For each iteration in SGD, parameters are updated by the gradient of the loss for one random sample.

For example, the loss for a sample $(u, i) \in S$ in the equation (2.3) is

$$
l(p_u, q_i) = \frac{1}{2} \{(r_{ui} - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2)\}.
$$

Then the gradient of the loss for $(u, i)$ is calculated as

$$
\frac{\partial l}{\partial p_u} = -(r_{ui} - p_u^T q_i) q_i + \lambda p_u,
$$

$$
\frac{\partial l}{\partial q_i} = -(r_{ui} - p_u^T q_i) p_u + \lambda q_i.
$$

Using the notation of inner product $\langle x, y \rangle = x^T y$, we summarize the algorithm of SGD for MF as Algorithm [1]
Algorithm 1 SGD algorithm for MF
1: Initialize $p_u$s, $q_i$s and set the step size $\eta$ and regularization parameter $\lambda$
2: while not converged do
3: Shuffle the samples randomly
4: for each sample $(u, i) \in S$ do
5: update $p_u$ and $q_i$ as
6: $p_u^{\text{new}} = p_u^{\text{old}} + \eta \cdot \{(r_{ui} - \langle p_u^{\text{old}}, q_i^{\text{old}} \rangle)q_i^{\text{old}} - \lambda p_u^{\text{old}}\}$
7: $q_i^{\text{new}} = q_i^{\text{old}} + \eta \cdot \{(r_{ui} - \langle p_u^{\text{old}}, q_i^{\text{old}} \rangle)p_u^{\text{old}} - \lambda q_i^{\text{old}}\}$
8: end for
9: end while

2.3 Matrix Factorization for Implicit Feedback

In this section we explain how Hu, Koren, and Volinsky (2008) applied MF to implicit feedback. From now on, $r_{ui}$ is implicit feedback, such as the number of views or clicks. The simplest approach for implicit feedback is to binarize it then regard as ‘preference’ and just apply the existing methods for explicit feedback to this binarized feedback. In detail, let’s consider the binary variable $y_{ui}$ which indicates the preference of user $u$ about item $i$. $y_{ui}$ is defined as

$$y_{ui} = \begin{cases} 
1 & r_{ui} > 0 \\
0 & r_{ui} = 0
\end{cases}.$$

One can just apply MF to the rating matrix consisting of $y_{ui}$. However, this approach would be dangerous as assuming all implicit feedback is positive at same level.

As we talked about implicit feedback in section 2.1, the numerical value of implicit feedback implies confidence and Hu, Koren, and Volinsky (2008) adapted MF by introducing confidence $c_{ui}$. Confidence $c_{ui}$ is defined as an
increasing function of implicit feedback $r_{ui}$, such as

$$c_{ui} = 1 + \alpha r_{ui} \quad \text{or} \quad c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon).$$

(2.5)

And the loss function is adapted as the weighted sum of squared errors where the weight is $c_{ui}$.

$$L(P, Q) = \frac{1}{2} \sum_{(u, i) \in U \times I} c_{ui}(y_{ui} - p_u^T q_i)^2 + \lambda(\|P\|^2 + \|Q\|^2) \quad (2.6)$$

Note that the loss is calculated for all $(u, i) \in U \times I$ since we set $y_{ui} = 0$ for all unobserved $(u, i)$. For more details, see Hu, Koren, and Volinsky (2008).

In fact, minimizing the loss $L(P, Q)$ without the penalty term is exactly the same with maximizing the log-likelihood of Gaussian model, $y_{ui} \sim N(p_u^T q_i, c_{ui}^{-1})$. Therefore, the variance of the preference $y_{ui}$ is $c_{ui}^{-1}$ and the meaning of confidence $c_{ui}$ is clear. By introducing the concept of confidence $c_{ui}$, $y_{ui}$ with the higher $c_{ui}$ is more important in obtaining the parameters $P, Q$. This approach was also used in Johnson (2014), and we use the same approach to build our model in chapter 3.

### 2.4 Bayesian Personalized Ranking (BPR)

In this section we summarize the Bayesian personalized ranking introduced by Rendle et al. (2009). BPR is a recommendation framework for implicit feedback based on Bayesian approach. BPR assumes that there are user-specific pairwise preferences, so that the user can order all items. Also, it is assumed that users prefer items they consumed than others.

In formal, it is assumed that for each user $u \in U$ there is user-specific total order $\geq_u$ of all items. For $i, j \in I$, $i \geq_u j$ means that user $u$ prefers $i$ than $j$. Ordering $\geq_u$ satisfies the following statements:
For all $i, j, k \in I$ and $u \in U$,

$i \geq_u j$ or $j \geq_u i$ \hspace{1cm} (totality),

If $i \geq_u j$ and $j \geq_u i$ then $i = j$ \hspace{1cm} (antisymmetry), \hspace{1cm} (2.7)

If $i \geq_u j$ and $j \geq_u k$ then $i \geq_u k$ \hspace{1cm} (transitivity).

In the setting of BPR, $(i \geq_u j)$ is an event in probabilistic perspective. Therefore, the statements in (2.7) are written as follows:

For all $i, j, k \in I$ and $u \in U$,

$Pr(i \geq_u j$ or $j \geq_u i) = 1$ \hspace{1cm} (totality),

If $i \neq j$ then $(i \geq_u j) \cap (j \geq_u i) = \emptyset$ \hspace{1cm} (antisymmetry), \hspace{1cm} (2.8)

$(i \geq_u j) \cap (j \geq_u k) \subseteq (i \geq_u k)$ \hspace{1cm} (transitivity).

Let’s define random variables $X_{uij} := I(i \geq_u j)$, $X_{uij} \sim Bernoulli(p_{uij})$ for all $u \in U$, $i, j \in I$, $i \neq j$, where $p_{uij} = Pr(i \geq_u j)$. By the totality and antisymmetry in (2.8) it is easily shown that

$X_{uij} + X_{uji} = 1$ and $p_{uij} + p_{uji} = 1 \hspace{1cm} \forall u \in U, i, j \in I, i \neq j$.

Rendle et al. \cite{2009} modeled the probability of user $u$ prefer item $i$ than $j$ by some scores $y_{ui}$, $y_{uj}$ as

$Pr(i \geq_u j) = \sigma(y_{ui} - y_{uj}) \hspace{1cm} (2.9)$

where $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function and $y_{ui}$ is the score of $(u, i) \in U \times I$. The score $y_{ui}$ is modeled by any other explicit implicit methods, usually MF in the equation (2.2). Note that the modeling in the equation (2.9) satisfies

$Pr(i \geq_u j) + Pr(j \geq_u i) = 1$

since $\sigma(x) + \sigma(-x) = 1$ so that it doesn’t conflict with the totality and antisymmetry in (2.8).
It is believed that users prefer items which they consumed than others. Therefore, with the additional notations \( I_u^+ := \{ i \in I : (u, i) \in S \} \), \( I_u^- := I \setminus I_u^+ \), we have observations as

\[
i \geq_u j \quad \text{for all } u \in U, \ i \in I_u^+, \ j \in I_u^- . \tag{2.10}\]

We define the set of \((u, i, j)\) as

\[
D_S := \{ (u, i, j) \in U \times I \times I : u \in U, \ i \in I_u^+, \ j \in I_u^- \}. \]

Then, the observations in (2.10) are expressed as

\[
X_{uij} = 1 \quad \text{for all } (u, i, j) \in D_S.
\]

With the assumption that \(X_{uij}\) for all \((u, i, j) \in D_S\) are independent, the likelihood is written as

\[
\prod_{(u, i, j) \in D_S} p_{uij}^{x_{uij}} (1 - p_{uij})^{1-x_{uij}}
\]

\[
= \prod_{(u, i, j) \in D_S} p_{uij} \quad (\because x_{uij} = 1 \text{ for } (u, i, j) \in D_S)
\]

With MF in the equation (2.2) for the score \(y_{ui}\) and the modeling in the equation (2.9), \(y_{ui} = b_i + p_u^T q_i\), and the log-likelihood is written as

\[
\sum_{(u, i, j) \in D_S} \log \sigma(y_{ui} - y_{uj})
\]

\[
= \sum_{(u, i, j) \in D_S} \log \sigma(b_i - b_j + p_u^T (q_i - q_j))
\]

Note that \(\mu, b_u\) are not needed since they don’t affect \(y_{ui} - y_{uj}\). The loss function \(L(\theta)\) of BPR, so-called BPR-OPT criterion in Rendle et al. (2009),

\[
L(\theta) := \sum_{(u, i, j) \in D_S} -\log \sigma(b_i - b_j + p_u^T (q_i - q_j)) + R_{uij}(\theta), \tag{2.11}
\]
where $R_{uij}(\theta) = \frac{1}{2} \lambda (\|b_i\|^2 + \|b_j\|^2 + \|p_u\|^2 + \|q_i\|^2 + \|q_j\|^2)$ is the regularization term, is not different from the negative log-likelihood with $\ell_2$-regularization. Therefore, the estimator $\hat{\theta}$ of parameters is MLE with regularization.

Let $l_{uij}(\theta)$ be the loss for one sample $(u, i, j) \in D_S$, then $L(\theta) = \sum_{(u, i, j) \in D_S} l_{uij}(\theta)$.

Note that all parameters are regularized with same $\lambda$ in the equation (2.11), however, it is possible to regularize parameters with different levels of $\lambda$, e.g., $\nabla l_{uij}(\theta)$ is obtained with different levels of $\lambda$ as follows:

\[
\begin{align*}
\frac{\partial l_{uij}(\theta)}{\partial b_i} & = -(1 - \sigma(y_{ui} - y_{uj})) + \lambda_b b_i \\
\frac{\partial l_{uij}(\theta)}{\partial b_j} & = (1 - \sigma(y_{ui} - y_{uj})) + \lambda_b b_j \\
\frac{\partial l_{uij}(\theta)}{\partial p_u} & = -(1 - \sigma(y_{ui} - y_{uj})) (q_i - q_j) + \lambda_{uf} p_u \\
\frac{\partial l_{uij}(\theta)}{\partial q_i} & = -(1 - \sigma(y_{ui} - y_{uj})) p_u + \lambda_+ q_i \\
\frac{\partial l_{uij}(\theta)}{\partial q_j} & = (1 - \sigma(y_{ui} - y_{uj})) p_u + \lambda_- q_j
\end{align*}
\]

where $\lambda_b, \lambda_{uf}, \lambda_+, \lambda_-$ are regularization parameters for item bias, user factor, positive item factor, negative item factor respectively.

SGD updating equations are as follows:

\[
\begin{align*}
b_i^{\text{new}} & = b_i^{\text{old}} - \eta \{ -(1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}})) + \lambda_b b_i^{\text{old}} \} \\
b_j^{\text{new}} & = b_j^{\text{old}} - \eta \{ (1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}})) + \lambda_b b_j^{\text{old}} \} \\
p_u^{\text{new}} & = p_u^{\text{old}} - \eta \{ -(1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}))(q_i^{\text{old}} - q_j^{\text{old}}) + \lambda_{uf} p_u^{\text{old}} \} \\
q_i^{\text{new}} & = q_i^{\text{old}} - \eta \{ -(1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}})) p_u^{\text{old}} + \lambda_+ q_i^{\text{old}} \} \\
q_j^{\text{new}} & = q_j^{\text{old}} - \eta \{ (1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}})) p_u^{\text{old}} + \lambda_- q_j^{\text{old}} \} 
\end{align*}
\] (2.12)

Basic SGD algorithm shuffles all observations and training them, but Rendle et al. (2009) sampled one sample at each iteration since $|D_S|$ is too large. For more discussion about sampling methods, see Rendle and Freudenthaler (2014).
We summarize this section by describing SGD algorithm of BPR with uniform sampling as Algorithm 2.

**Algorithm 2** SGD algorithm for BPR with MF

1. Initialize $b_i$, $p_u$, $q_i$ and set $\eta$, $\lambda_b, \lambda_u, \lambda_+, \lambda_-$
2. while not converged do
3. Uniformly sample $(u, i, j) \in D_S$
4. update $b_i, b_j, p_u, q_i, q_j$ as the updating equations in (2.12)
5. end while
Chapter 3

Bayesian Personalized Ranking with Count Data

In this chapter, we propose a variation of BPR model handling the non-binary data. Remind that The original BPR considers only the binary data. Therefore, it doesn’t use entire implicit feedback. We borrow the concept of confidence in Hu, Koren, and Volinsky (2008) to reflect the difference in numeric values of implicit feedback. After the explanation, we compare the implementation result of our model with BPR.

3.1 Notation

Before explaining our model, we redefine some notations which almost same with notations in chapter 2. Let $U$ is the set of all users, $I$ is the set of all items. $r_{ui}$ is the user $u$’s non-binary implicit feedback about item $i$, e.g., it may be the count of views. $R = (r_{ui})_{(u,i) \in U \times I}$ is the $|U| \times |I|$ matrix of implicit feedback. $S := \{(u,i) \in U \times I : r_{ui} \text{ is observed}\}$ is the set of user-item pairs.
whose related rating is observed. \( I_u^+ := \{ i \in I : (u, i) \in S \} \) is the set of items which user \( u \) consumed, and \( I_u^- := I \setminus I_u^+ \) is the set of other items. We call \( I_u^+, I_u^- \) as positive items and negative items, respectively. Finally, we define \( D_S := \{(u, i, j) \in U \times I \times I : i \in I_u^+, j \in I_u^-\} \).

### 3.2 Model

In this section, we suggest the loss function and explain the model in detail. As Hu, Koren, and Volinsky (2008) said and we explain in section 2.3, the numeric value of implicit feedback implies confidence level. In order to consider differences of the numeric values of implicit feedback, we used the same approach in Hu, Koren, and Volinsky (2008) and Johnson (2014). Define the confidence \( c_{ui} \) as the increasing function of \( r_{ui} \). In this paper, we used

\[
   c_{ui} = \alpha (1 + \log r_{ui}).
\]

In most cases \( r_{ui} > 0 \) or \( r_{ui} \geq 1 \) for \( (u, i) \in S \) so \( \log r_{ui} \) is well-defined. Using the same notation in section 2.4, let \( p_{uij} = Pr(i \geq_u j) \). We suggest the loss function of the form:

\[
   L(\theta) := \sum_{(u,i,j) \in D_S} -c_{ui} \log p_{uij} + R_{uij}(\theta). \tag{3.2}
\]

When we use MF to modeling the score, the loss function is written as

\[
   L(\theta) := \sum_{(u,i,j) \in D_S} -c_{ui} \log \sigma(b_i - b_j + p_u^T(q_i - q_j)) + R_{uij}(\theta) \\
   := \sum_{(u,i,j) \in D_S} l_{uij}(\theta) \tag{3.3}
\]

where \( R_{uij}(\theta) = \frac{1}{2} \{ \lambda_b(\|b_i\|^2 + \|b_j\|^2) + \lambda_u\|p_u\|^2 + \lambda_+\|q_i\|^2 + \lambda_-\|q_j\|^2 \} \).
3.3 Meaning of $c_{ui}$

We explain the role of confidence $c_{ui}$ in our model. Remind that the loss function and the likelihood of BPR are

$$\sum_{(u,i,j)\in D_S} -\log p_{uij} + R_{uij}(\theta),$$

$$\prod_{(u,i,j)\in D_S} p_{uij}.$$  

Therefore, our loss in the equation (3.2) is just weighted form of loss in original BPR,

$$\sum_{(u,i,j)\in D_S} -c_{ui} \log p_{uij} + R_{uij}(\theta),$$

where the weights are confidences $c_{ui}$. To minimize this, it is more effective to minimize $-\log p_{uij}$ with the higher $c_{ui}$ than lower one. Therefore, our model reflects the difference of implicit feedback as the confidence.

In the perspective of probability, our loss function in the equation (3.2) implies that the likelihood is

$$\prod_{(u,i,j)\in D_S} p_{uij}^{c_{ui}} = \prod_{u\in U} \prod_{i\in I_u^+} \prod_{j\in I_u^-} p_{uij}^{c_{ui}}$$

$$= \prod_{u\in U} \prod_{i\in I_u^+} \left( \prod_{j\in I_u^-} p_{uij} \right)^{c_{ui}}$$

$$= \prod_{u\in U} \prod_{i\in I_u^+} Pr(i \geq u, \forall j \in I_u^-)^{c_{ui}}$$ (3.4)

Let’s assume that $c_{ui} \in \mathbb{N}$ and define the random variables

$$X_{ui} \sim Binomial(c_{ui}, p_{ui}),$$

where $p_{ui} := Pr(i \geq u, \forall j \in I_u^-)$. Then the equation (3.4) implies that our likelihood is probability that we observe $X_{ui} = c_{ui}$ for $u \in U, i \in I_u^+$. Therefore,
\( c_{ui} \) is considered as the number of observations that user \( u \) prefer item \( i \) than all negative items \( j \in I_u^- \). In this sense, definition of \( c_{ui} \), the increasing function of \( r_{ui} \), is natural since the numeric value of implicit feedback \( r_{ui} \) means frequency or count in almost cases. Although \( c_{ui} \notin \mathbb{N} \) in the equation (3.1) breaks the binomial structure, meaning of \( c_{ui} \) is not different from what we have explained.

There are related studies which consider the weight or confidence. Gantner et al. (2012) considers weights on the loss of BPR. However, they don’t adapt the loss in direct. Instead, they adapt the sampling probability in SGD. Because of it, the updating equation of WBPR in Gantner et al. (2012) actually minimizes different loss:

\[
\sum_{(u,i,j) \in D_S} -w_{uij} \log p_{uij} + w_{uij} R_{uij}(\theta)
\]

where \( R_{uij} \) is the regularization term for one sample \((u, i, j)\). But, our model minimizes

\[
\sum_{(u,i,j) \in D_S} -w_{uij} \log p_{uij} + R_{uij}(\theta)
\]

where \( w_{uij} = c_{ui} \). Also, the intention of regarding \( w_{uij} \) is different from us. Their goal is to apply different weights to the loss when the data is sampled non-uniformly.

Wang et al. (2012) adapt BPR by regarding the confidence. Their loss function is

\[
\sum_{(u,i,j) \in D_S} -\log \sigma(c_{uij}(y_{ui} - y_{uj})) + R_{uij}(\theta).
\]

It is unclear what is the role of \( c_{uij} \) actually in probabilistic perspective. Since \( c_{uij} = -c_{uji} \) in Wang et al. (2012),

\[
Pr(i \geq u j) = \sigma(c_{uij}(y_{ui} - y_{uj})) = \sigma(c_{uji}(y_{uj} - y_{ui})) = Pr(j \geq u i)
\]
So, \( Pr(i \geq u, j) + Pr(j \geq u, i) \neq 1 \) and \( \geq u \) doesn’t satisfy the definition of total order in 2.8. In the other hand, the meaning of \( c_{ui} \) in our model is clear and we show it in the next section.

### 3.4 SGD Algorithm

In this section we describe the SGD algorithm for minimizing the loss function in the equation (3.3). Because the loss function which we proposed is similar with the loss of original BPR in the equation (2.11), updating equation is almost same.

\[
\begin{align*}
    b_i^{\text{new}} &= b_i^{\text{old}} - \eta \{ -c_{ui} \left( 1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}) \right) + \lambda b_i^{\text{old}} \} \\
    b_j^{\text{new}} &= b_j^{\text{old}} - \eta \{ c_{ui} \left( 1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}) \right) + \lambda b_j^{\text{old}} \} \\
    p_u^{\text{new}} &= p_u^{\text{old}} - \eta \{ -c_{ui} \left( 1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}) \right) (q_i^{\text{old}} - q_j^{\text{old}}) + \lambda u f p_u^{\text{old}} \} \\
    q_i^{\text{new}} &= q_i^{\text{old}} - \eta \{ -c_{ui} \left( 1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}) \right) p_u^{\text{old}} + \lambda_+ q_i^{\text{old}} \} \\
    q_j^{\text{new}} &= q_j^{\text{old}} - \eta \{ c_{ui} \left( 1 - \sigma(y_{ui}^{\text{old}} - y_{uj}^{\text{old}}) \right) p_u^{\text{old}} + \lambda_- q_j^{\text{old}} \}
\end{align*}
\]

(3.5)

And the SGD algorithm is also almost same with Algorithm 2.

**Algorithm 3** SGD algorithm for minimizing the loss in the equation (3.3)

1: Initialize \( b_i \)'s, \( p_u \)'s, \( q_i \)'s, set \( \eta, \lambda_b, \lambda_u f, \lambda_+, \lambda_-, \alpha \), calculate \( c_{ui} \)'s

2: **while** not converged **do**

3:    Uniformly sample \((u, i, j) \in D_S\)

4:    update \( b_i, b_j, p_u, q_i, q_j \) as the updating equations in (3.5)

5: **end while**
3.5 Implementation

In order to implement our model and BPR, we use Steam video games data which the Tamber team formatted in Kaggle. Steam is the most popular PC gaming hub. And, the data contains the Steam user’s interactions with video games. This data consists of user-id, game-title, behavior-name, value. The behaviors are ‘play’ and ‘purchase’. In the case of ‘play’ the value means playtime which is the number of hours the user has played the game. More details are available on the website https://www.kaggle.com/tamber/steam-video-games. We use only the ‘play’ and playtime data, not ‘purchase’ data. We use \( r_{ui} \) as the rounded value of playtime. Also, We consider only the users who have played more than one game to avoid cold-start situation. Table 3.1 describes the configuration of Steam Video Games data after preprocessing.

<table>
<thead>
<tr>
<th># of Users</th>
<th># of Games</th>
<th># of Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,310</td>
<td>3,217</td>
<td>53,294</td>
</tr>
</tbody>
</table>

Table 3.1: Configuration of Steam Video Games data after preprocessing

In order to compare our model with BPR, we randomly sample one feedback from each user who has played more than five games. Then, we set the sampled data as test data and the others as training data. Table 3.2 and Table 3.3 describe configuration of training and test data, respectively.

<table>
<thead>
<tr>
<th># of Users</th>
<th># of Games</th>
<th># of Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,310</td>
<td>3,217</td>
<td>51,430</td>
</tr>
</tbody>
</table>

Table 3.2: Configuration of training data
Figure 3.1: Histogram of playtime. The number of users who played for hours of playtime is exponentially decreasing as the playtime increasing.

Figure 3.1 is the histogram of playtime. In this histogram we can see that the number of users who played for hours of playtime is exponentially decreasing. In order to reduce the influence of scaling, we use logarithm to obtain $c_{ui}$ like the equation (3.1).

We use area under the ROC curve (AUC) as the metric of accuracy to compare models, where receiver operating characteristic (ROC) curve is the plot of the true positive rate against the false positive rate at various thresholds.
In the setting of BPR, we can define AUC per user and AUC as

\[
AUC(u) := \frac{1}{|I_u^+||I_u^-|} \sum_{i \in I_u^+} \sum_{j \in I_u^-} I(\hat{y}_{ui} - \hat{y}_{uj} > 0)
\]

\[
AUC := \frac{1}{|U|} \sum_{u \in U} AUC(u)
\]

(3.6)

Table 3.4, Table 3.5 and Figure 3.2 show the results of implementation. Note that here one epoch means the number of feedbacks \(|S|\). We set the hyper-parameters \(k = 10, \eta = 0.1, \alpha = 0.3, \lambda_b = 0.5, \lambda_{uf} = 0.0025, \lambda_+ = 0.0025, \lambda_- = 0.000025\). For initialization, we set \(b_i\) as zero, and components of latent factors are randomly sampled from \(N(0, 0.01)\).

In order to compare the losses of both models, \(\alpha = 0.3\) is obtained to make the scales of both losses be similar at the start of training. In detail, it can be assumed that all \(p_{uij} \simeq 0.5\) at the start by the initialization and \(|I| \gg |I_u^+|\) so that \(|I_u^-| \simeq |I|\) for all \(u \in U\). Then,

\[
\sum_{(u,i,j) \in D_S} c_{ui} \log p_{uij} \simeq \log 0.5 \sum_{(u,i,j) \in D_S} c_{ui}
\]

\[
= \log 0.5 \sum_{(u,i) \in S} c_{ui} |I_u^-|
\]

\[
\simeq |I| \log 0.5 \sum_{(u,i) \in S} c_{ui}
\]

Similarly, \(\sum_{(u,i,j) \in D_S} \log p_{uij} \simeq |I| \log 0.5 \sum_{(u,i) \in S} 1\). To make the both similar, we set \(\alpha\) as

\[
\frac{1}{|S|} \sum_{(u,i) \in S} c_{ui} \simeq 1 \Leftrightarrow \alpha \simeq |S| \left( \sum_{(u,i) \in S} (1 + \log r_{ui}) \right)^{-1}
\]
Table 3.4: Maximum test AUC of our model (CBPR) and BPR

<table>
<thead>
<tr>
<th></th>
<th>CBPR</th>
<th>BPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum test AUC</td>
<td>0.9343</td>
<td>0.9308</td>
</tr>
</tbody>
</table>

Table 3.5: Test AUC of our model (CBPR) and BPR when each test loss is minimized

<table>
<thead>
<tr>
<th></th>
<th>CBPR</th>
<th>BPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test AUC when test loss is minimized</td>
<td>0.9343</td>
<td>0.9292</td>
</tr>
</tbody>
</table>

Figure 3.2: Plots about loss and AUC of BPR and our model (CBPR). ‘train’ and ‘test’ indicates the result for training data and test data, respectively.
The results in Table 3.4 and 3.5 show that test AUC of our model is better than BPR. In Figure 3.2, the training AUC of BPR is higher than our model but the difference is decreasing, and the test AUC of our model is higher than BPR after enough epochs. Also, the training loss of our model and BPR are almost same but test loss of our model is lower than BPR.
Chapter 4

Discussion

Although we have shown that our model works well in section 3.5, there are two issues to discuss. First, training losses of both model are almost same but our model has lower test loss. As we said, we set $\alpha$ so that the scales of losses are similar. Therefore, the similarity of both losses at the start of training is intended, however, the similarity at overall iteration is not. Besides, the test loss of our model is remarkably lower than BPR after enough iterations. It can be interpreted as our model is better at preventing the overfitting problem than BPR.

Second, test AUC of our model is maximized when test loss is minimized. Rendle et al. (2009) emphasized that one of the strengths of BPR is that minimizing the loss function of BPR is approximately same with maximizing AUC. In this sense, we guess that our model is more closer to maximize AUC than BPR. In order to investigate these features, further works will be required.
References


국문초록

베이지안 개인화 순위(Bayesian personalized ranking; BPR) 모형은 내재적 피드백을 이용하는 뛰어난 모형 중 하나이다. 그러나, 기존의 BPR 모형은 이진 형태의 내재적 피드백만을 다룬다는 한계점이 있다. 이 논문에서는 이 한계점을 극복하기 위해, 횟수 자료(count data)와 같은 내재적 피드백의 수치값을 고려한 변형된 형태의 BPR 모형을 제시한다. 더 나아가, 우리의 모형과 기존의 BPR을 R에서 구현하고 결과를 비교한다. 이 모형은 BPR 보다 내재적 피드백을 강하게 반영하는데 유용할 것으로 생각된다.

주요어: 추천 시스템, 내재적 피드백, 횟수 자료, 행렬분해, 베이지안 개인화 순위(BPR).

학번 : 2015-22579