Improving Customized Recommendation Accuracy Including User’s Expectancy in Bandwagon Phenomenon

Lu Sun 1 and Kiejin Park 2

1Postgraduate, Department of Industrial Engineering, Ajou University, Suwon, South Korea.

2Professor, Department of Integrative Systems Engineering, Ajou University, Suwon, South Korea.

2Orcid: 0000-0002-6872-2806 & Scopus Author ID: 55654521300

Abstract

With the number of users and items has grown rapidly in online, recommender systems have come to play an important role in terms of helping users to choose items. In the process of consumer decision making, user tend to follow other people’s opinions as a way to reduce the risk of making bad decision which is known as bandwagon effect. So if users see an item with high popularity that means that the other people also like the item, they will have a look forward to the item which can be seen as users’ expectancy. However, very few studies have considered the impact of the expectancy on the users’ evaluation of items. In this paper, we not only define user’s expectancy but also standardize it by considering different users have different rating standards. Then we have developed a recommendation system that considers the psychological concept of bandwagon effect. As a result, our mechanism outperforms the existing ALS (Alternating Least Square) model in improving the prediction accuracy on RMSE.

Keywords: Recommendation, Expectancy, Bandwagon Phenomenon, Apache Spark, Hadoop

INTRODUCTION

In recent years, with the rapid development and the widespread of internet, users’ life is full of a large number of choices. Electronic retailers and product providers offer a variety of products and services to meet the different needs of different users, allowing users to choose items according to their interests and other people’s opinions. But in the face of vast amounts of information, it is very hard for users to quickly choose items. Recommender systems is an effective way to solve this contradiction and can help users solve the problem of information overload.

Moreover more and more scholars of recommender systems have focused on the Bandwagon Effect a kind of psychological phenomenon that a person’s attitudes are influenced by other people [1]. This effect has been confirmed in recommender systems through various studies. S. Shyam et. al. described the information as product ratings and sales rank influenced user behavior on Amazon.com-a famous e-commerce websites [2]. Then they found that images and metrics conveying others’ opinions and expert opinions are psychologically significant to users [3]. In 2015, Choi et al. proved the bandwagon effect existed in movie recommendation domain [4]. And this effect is also applied to improvement the performance of recommender systems in other studies especially in recent years. Y. Chou et. al. demonstrated two novel product recommendation approaches based on the bandwagon effect of virtual houses in virtual worlds [5]. S. Kang showed a new design on collaborative filtering recommendation system using bandwagon effect [6]. However they did not take an analysis of user’s expectancy caused by the bandwagon effect and the effect of user’s rating standard in quantifying the expectancy into account. These two important factors presented in this paper according to that a user’s shopping behavior or preference for certain items could be influence by the bandwagon effect as follows: the first is the user’s expectancy caused by bandwagon effect, and the second is the rating standard in users’ rating process. The user’s expectancy refers to when a user’s preferences to an item are influenced by other people who have rated or bought the item, that is, before he/she purchase the item, other people’s opinions will give he/she an impression and he/she will have an expectation of the item in heart. The second factor refers to the fact that users have their own rating standard, for example, the average rating of an item is 3 with a maximum of 5 meaning like the most, to some user this is a good rating but to the others this rating is not good so this item will give different users different expectations. Besides, we explore our experiments on a real large scale dataset on Hadoop-Spark framework which can not only process big data but also can make high-speed operations [7].

The rest of the paper is organized as follows: Chapter 2 contains a brief introduction of the related work, and Chapter 3 describes the process of our model, Chapter 4 shows the performance evaluations of the proposed model, in the end we presents the conclusions in Chapter 5.
RELATED RESEARCH

Matrix Factorization and Alternating Least Square (ALS)

In recommender system, rating prediction is one of the most important tasks. To accurately solve the task, one must model the characteristic of both users and items precisely. Matrix Factorization (MF) as a latent factor model in collaborative filtering is the most widely used approach for this task. It represents users and items in a latent factor space of dimensionality $f$—each item $i$ is associated with a vector $q_i \in \mathbb{R}^f$, and each user $u$ is associated with a vector $p_u \in \mathbb{R}^f$. The rating prediction of user $u$ to item $i$ with the dot product $q_i^T p_u$ is denoted by $r_{u,i}$, leading to the estimate \[ \hat{r}_{u,i} = q_i^T p_u \] (1)

To learn the feature vectors $q_i$ and $p_u$ the system minimizes the regularized squared error on the set of known ratings:

\[
\min_{p^*,q^*} \sum_{u,i \in \mathcal{K}} (r_{u,i} - q_i^T p_u)^2
\] (2)

ALS (Alternating Least Square) has been mostly utilized to address the minimization problem because of its programming simplicity \[9\]. And it optimizes the MF by finding optimal factor weights to minimize the least squares between predicted and actual ratings in recommender systems.

Hadoop-Spark Framework

Spark is an in-memory processing framework that, in the case of iterative processing, loads data into memory and drives high-speed data analysis. Through this framework, it can solve the disk bottleneck caused by MapReduce, which is a large-scale data processing method in Hadoop, and can efficiently execute data processing requiring iterative processing such as machine learning \[7\]. In addition, Hadoop is a cluster system composed of a number of slave nodes for storing and processing data and a single master node for managing them, which can store and process large amounts of data. In Fig. 1, YARN (Yet Another Resource Negotiator) manages the entire resources, tasks, scheduling and defects of the Hadoop Cluster and parallelizes the work on each slave node through the Spark Driver Program of the Master Node. In this paper the prototype was constructed for performance evaluation of the proposed method based on the Hadoop-Spark framework.

Figure 1: The Cluster System Architecture of Hadoop-Spark Framework

Model Formulation: Customized Recommendation with User’s Expectancy

Bandwagon effect means a phenomenon that an item is popular with most of the people. By the other word, if an item with high popularity, user’s expectation to this item will also high. So user’s rating is not only influenced by his own preference but also affected by the expectation caused by the bandwagon effect. As a result, this paper aims to improve the recommendation accuracy of rating prediction by using the psychological phenomenon according to that the user’ expectancy to the item will be impact on the item’s popularity causing the bandwagon effect.

In online environment, the bandwagon effect cues maybe users’ reviews or items’ sales volume, users will imitate other people according to these cues. For example, if all of the other people are interested in one item so I will also show some interest in it even if actually I do not like it so much. According to Choi, they considered the bandwagon effect in movie recommendation domain and observed that the number of views is a more suitable factor to identify the popular movies than ratings \[4\]. So in our model, we use the number of views to measure items’ popularity degree as in equation (3),

\[
bw_{u,i} = \frac{Freq(f, i)}{H}
\] (3)

Here, $H$ is the total number of users before user $u$, $Freq(f, i)$ is the number of users who viewed item $i$ before user $u$.

In addition, by the bandwagon effect. We can easily know that if we see an item with high popularity and good reputation, we will generate a high expectancy to the item. For example, knowing that there are 10 people, and 8 people choose the item and rate the item with 4 (5 is being the best), the item will give us a good image and our expectancy to the item caused by the bandwagon effect will also react on our rating to the item. We quantify this expectancy in our model as in equation (4),
\[ e_{u,i} = \frac{bw_{u,i} * (SO_{u,i} - \bar{R}_u)}{R_u} \]  

(4)

Where \( bw_{u,i} \) is the bandwagon effect defined in equation (3), \( SO_{u,i} \) is item \( i \)'s average rating before user \( u \) rates item \( i \), \( \bar{R}_u \) is the user average rating.

In equation (4), the user average rating is taken into account because that in reality each user has own preference and has own standard for rating too. Some user’s rating is strict and low but the other are not. For instance, some people take an item with 3 average rating (5 being the best) as a good item but the other people may not agree that.

The new rating obtained by reflecting the obtained expectation to the item by each user is derived as shown in equation (5),

\[ p_{u,i} = r_{u,i} + \frac{e_{u,i} - E_{min}}{E_{max} - E_{min}} \]  

(5)

In the case, \( p_{u,i} \) is a score that reflects the expectation, \( r_{u,i} \) is the original evaluation score and \( E_{min} \) refers to the minimum and the \( E_{max} \) refers to maximum value of the expectation caused by the bandwagon effect.

**EXPERIMENTAL RESULTS**

The experimental cluster environment consists of 1 master and 6 slaves, and detailed cluster specification is shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1: Experimental Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Master</strong></td>
</tr>
<tr>
<td>CPU</td>
</tr>
<tr>
<td>Memory Size (Total)</td>
</tr>
<tr>
<td>Storage Size (Total)</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Spark Version</td>
</tr>
<tr>
<td>Hadoop Version</td>
</tr>
</tbody>
</table>

**Dataset**

We test the proposed method on GroupLens movie dataset. The dataset is designed for recommended algorithm research and testing. The dataset publishes user ratings along with user id and movie id which is composed of 26,000,000 ratings, for 45,000 movies by 270,000 users as shown in Table 2. The ratings of users to movies from 0.5–5 (with 5 being the best). The data set is divided randomly into M parts (M = 8 in this paper), one is chosen as the test set, and the rest M-1 is used as the training set.

<table>
<thead>
<tr>
<th>Table 2: Shows the Sample of GroupLens Movie Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>userId</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>10132</td>
</tr>
<tr>
<td>6467</td>
</tr>
</tbody>
</table>

**Evaluation Metrics**

In order to measure the prediction accuracy of our recommendation model, root mean square error (RMSE) which is the most important representatives of predictive accuracy metric [10], defined by the following:

\[ RMSE = \sqrt{\frac{(\hat{r}_{u,i} - r_{u,i})^2}{N}} \]  

(6)

In the above equation (6), \( N \) denotes the number of rating-prediction pairs in the test set, \( \hat{r}_{u,i} \) is a predicted rating and \( r_{u,i} \) is a real rating.

**RESULTS OF EXPERIMENT**

We verify our model under the Hadoop-Spark environment showed in Table 1. For the performance evaluation of this proposed model, the accuracy of the recommendation between the model (BW) applying the bandwagon effect and the Non-BW model was evaluated with RMSE. We considered item average rating and user average rating in our method so we also computed RMSE for these two models.

The RMSE values obtained using our approach and other models is presented in Fig. 2. The RMSE considering the bandwagon effect in our model is 0.8477. The metrics of Non-BW is 0.8510. We can observe that RMSE improves about 0.4% the bandwagon effect is taken into consideration although this change is not obvious. However, if we calculate RMSE only for ratings that are directly predicted by user/item’s average rating, we are easy to get that the metric greatly reduced form Fig.2.
CONCLUSION

In this paper, we model the bandwagon effect in online rating sites to understand the user rating behaviors and to improve the rating prediction accuracy. Analysis and experiment using large-scale data on Hadoop-Spark framework. Consequently, our model showed good accuracy for predicting personalized rating and has been proved superior to other models through experiments on a dataset in the movie domain. But there are still so many problems need to be solved in recommender system domain, such as cold-start, scalability issues, privacy issue. For further work we will focus on the study of more algorithms to solve the various challenges faced by the recommender system to improve the accuracy of the recommender system.

REFERENCES