Improving Top-K Contents Recommendation Performance by Considering Bandwagon Effect: Using Hadoop-Spark Framework

Suk-kyoon Kang[1] and Kiejin Park[2](✉)

[1] Department of Industrial Engineering, Ajou University, Suwon, Korea
tjrrbs1120@ajou.ac.kr
[2] Department of Integrative Systems Engineering, Ajou University,
Suwon, Korea
kiejin@ajou.ac.kr

Abstract. The study on the existing Collaborative filtering recommendation system is mainly aimed at improving the accuracy of prediction. However, in terms of actual recommendation service, it is more important that the Top-K recommendation list, which is effectively recommended to the user, is an item that the user actually likes, rather than improving the recommendation accuracy of all items. In this paper, we have developed a recommendation system that considers the psychological concept of Bandwagon Effect in order to improve the recommendation accuracy of the Top-K contents. For Big data distribution and storage, we used Hadoop and for the fast Big Data processing offering speed, we used Spark, an in-memory data processing framework for high-speed operations. As a result, the proposed model is superior to the existing model in terms of accuracy of recommendation for Top-K contents.

Keywords: Top-K contents · Bandwagon Effect · Apache Spark · Hadoop

1 Introduction

Collaborative filtering is a process in which connects the similar users each other based on the ratings of items, and a recommendation is made of items most suitable for users by predicting the rating of target users, So the key factor of this system is the accuracy of prediction. However, from the point of view of the recommendation service provider, it is more important to predict accurately of the preference for the Top-K items that actually recommend to user than predicting user's preference of all items accurately [1]. In this paper, we have developed a recommendation system that considers the psychological concept, Bandwagon Effect, in order to improve the recommendation accuracy of Top-K contents that will be effectively recommended. A prototype of the proposed model was developed by a fast, in-memory data processing framework Apache Spark, based on distributed storage system Hadoop. This paper consists of 5 sections, Sect. 2 describes the related research, and Sect. 3 gives the proposed recommendation system and its calculation process that considers the Bandwagon Effect. Section 4 shows the performance evaluation of the proposed model and lastly, the conclusion of the paper is given.

J. J. Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*, Lecture Notes in Electrical Engineering 474,
https://doi.org/10.1007/978-981-10-7605-3_23

2 Related Research

2.1 Matrix Factorization and ALS (Alternating Least Square)

Matrix factorization is a representative algorithm used in collaborative filtering. It decomposes a rating matrix into two feature vectors as shown in Eq. 1, and estimates a score vector of non-existing value by multiplying two feature vectors [2].

$$\min_{p,q} \sum_{u,i \in k} \left(r_{u,i} - p_u q_i\right)^2 \tag{1}$$

$r_{u,i}$ is a rating for item i of user u, and p_u and q_i are decomposed user feature vectors and item feature vectors. ALS (Alternating Least Square), which is one of the optimization techniques for finding more accurate feature vectors, has been adopted as an algorithm that meets the recent big data computing environment because it is operated in parallel, distributed form.

2.2 K-Means Clustering

K-Means Clustering is an algorithm for grouping given data into K clusters as shown in Eq. 2.

$$\min_S \sum_{i=1}^{k} \sum_{x \in S_i}^{n} ||x - \mu_i||^2 \tag{2}$$

This algorithm aims to find a set S that minimizes the sum of squares of the distance between the center point of each set in the given x individual data. And this algorithm is a representative unsupervised learning algorithm.

2.3 Bandwagon Effect

The Bandwagon Effect is a phenomenon that consumes the items that is popular among many people and many people support. Especially in recent years, as the use of products utilizing social network services, purchasing and sales have become active, the Bandwagon Effect effects on-line become prominent, and this phenomenon has been confirmed through various studies [3]. One study suggests that the user's choice and evaluation of movies can be used as an effective feature to improve the cold-start problem or the sparsity problem, which has been pointed out as a limit in CF recommendation system, suggesting the possibility of use [4]. However, in this study, they only mentioned the possibility as one of the features that can be applied to improve the limit of CF but did not develop a recommendation model. In our previous paper [5], we proposed a method using Bandwagon Effect, but in this paper, we did not consider the relative popularity when choosing popular items that cause Bandwagon Effect, only absolute criteria were considered. In this paper, we have developed a Top-K contents recommendation system that considers the practical Bandwagon Effect considering the relative popularity using the K-Means technique.

3 Suggested Model

3.1 Recommendation System Using the Bandwagon Effect

When users purchase items, they make a lot of choices affected by Bandwagon Effect, and if the recommendation system recommends a similar item with the popular item that causes this Bandwagon Effect, people naturally recall the popular item that they had purchased before and this leads to purchase. And in this paper, we call this phenomenon by ‘Expectancy’. As a result, this paper aims to improve the recommendation accuracy of Top-K contents by using the Psychological phenomenon that the anticipation raises by the Bandwagon Effect.

3.2 Selection of Popular Item

In order to consider the Bandwagon Effect, it is necessary to select a popular item that causes Bandwagon Effect. That is, in this paper, popular items were selected through two criteria, Sales Rate (s) and Reputation (r), in order to select popular items. First, in the case of the Sales Rate, many items sold have a high sales rate (Absolute Sales Rate), and items with high sales rates are often popular items. However, in reality, there are cases where items with high sales rates are not popular items. In other words, it is reasonable to measure the ‘relative Sales Rate’ within a group composed of users with similar preference rather than ‘absolute Sales Rate’. In this paper, considering the relativity of item’s popularity, we calculate the ratio of specific item in the overall group is compared with the proportion of specific item in the neighborhood group with similar preference as in Eq. 3.

$$S_i = \frac{I \cdot Freq(g,i)}{Freq(i) \cdot Freq(g)} \tag{3}$$

Here, I is the total number of items, $Freq(i)$ is the number of specific items, and $Freq(g,\ i)$ is the number of specific items in each group, and $Freq(g)$ is the number of items in each group and the Sales Rate of each item S_i is derived through the calculation between them. In addition, reputation (r) should be considered because popular products should not only have a high sales rate but also a good reputation. For that reason the reputation in this paper only considers items with high ratings (more than 4 points), and each item’s reputation is calculated as the ratio of high-rated items in the total number of items as in Eq. 4.

$$R_i = \frac{Freq(h_i)}{H} \tag{4}$$

Here, if H is the total number of high-rated items and $Freq(h_i)$ is the number of high-rated items of individual items, the reputation of each item (R_i) is derived through the calculation between them. Through the standardization process of these two criteria, the popularity coefficient (PC) of each item is defined as in Eq. 5. Then, based on the calculated PC value, a group of popular items is determined.

$$(PC)_i = S_i \times R_i (0 \leq PC \prec 1) \quad (5)$$

3.3 Expectancy

Expectancy is defined as Eq. 6, which calculates the similarity between a popular item group and other items, using the item feature vector extracted through Matrix Factorization technique based on the existing ratings.

$$e_{g,i} = \sim (i_g, i_j) = \frac{\sum_{k=1}^{n} \left(w_{g,k} \cdot w_{j,k}\right)}{\sqrt{\sum_{k=1}^{n} \left(w_{g,k}^2 \cdot w_{j,k}^2\right)}} \quad (6)$$

Here, $w_{g,k}$ is the item represents number of n feature vector k of the popular item group, $w_{j,k}$ is the number of n feature vector k of the other items, and the expectation of individual items $e_{g,i}$ is derived through calculation of similarity between them. The new rating obtained by reflecting the obtained expectation to the item purchased by each user is derived as shown in Eq. 7.

$$p_{u,i} = r_{u,i} + \frac{e_{u,i} - E_{\min}}{E_{\max} - E_{\min}} \quad (7)$$

In this case, $p_{u,i}$ is a rating that reflects the expectation, and $E_{\min}$ refers to the minimum and the $E_{\max}$ refers to maximum value of the expectation of the user, and ($e_{u,i}$) is the expected expectation of the specific user.

3.4 The Proposed Recommendation Process

The recommendation system that considers the Bandwagon Effect consists of two processes. (1) preprocessing step for extracting rating data considering the Bandwagon Effect and (2) constructing a recommendation system based on ALS algorithm using Hadoop-based Spark framework using preprocessed data.

4 Recommendation System Based on Spark Framework

4.1 Hadoop Based Spark Framework

Spark, a distributed processing system, its main driver program distributes each task to each slave node in a distributed-parallel manner. Spark uses the data processing structure of Directed Acyclic Graph (DAG) format called Resilient Distributed Datasets (RDD), which makes it possible to efficiently use the memory of all the slave nodes constituting the cluster. The cluster constructed for the experiment consists of one master node and six slave nodes, and the experimental environment of each node and datasets are shown in Table 1.

Table 1. Shows experimental environment and datasets

Specification		
	Master	Slave
Memory size (total)	16 GB	64 GB* 6(384 GB)
Storage size (total)	6 TB	8 TB *6 (48 TB)
OS	Ubuntu 16.04 LTS	Ubuntu 14.04 LTS
Spark version	2.01	2.01
Hadoop version	2.72	2.72
Dataset	Movie Lens Dataset *(consist of UserID, MovieID, Rating, Timestamp)* [6]	

4.2 Evaluation Metrics

In order to measure the accuracy of the recommendation model, RMSE and Precision@K were used as evaluation metric.

$$RMSE = \sqrt{\frac{(\hat{r}_{u,i} - r_{u,i})^2}{N}(r_{u,i} \geq 3)} \tag{8}$$

The RMSE (Root Mean Squared Error) of Eq. 8, which is the first evaluation metric, is a representative recommendation accuracy evaluation metric [7]. Precision@K (Precision at K) is a metric that evaluates the accuracy of the Top-K result lists as shown in Eq. 9, and this is widely used as evaluation metric in the IR (Information Retrieval) field and especially in ranking-based recommendation systems.

$$\mathrm{Pr}ecision@K = \frac{1}{M}(\sum_{i=0}^{M-1} \frac{1}{k} \sum_{j=0}^{\min(|D|,k)-1} rel_{D_i}(R_i(j)) \tag{9}$$

4.3 Result of Experiment

For the performance evaluation of this proposed model, the accuracy of the recommendation between the proposed model (BW) applying the Bandwagon Effect and the non-BW model was evaluated with RMSE and Precision@K. As shown in Table 2, the proposed model records about 10% more accurate recommendation for high-content content. In addition, the accuracy of Top-K content recommendation was superior to existing models that did not apply the top four proposed models, and the top 5–10 accuracy was superior to the existing models. From the standpoint of users, when users see the recommendation list, they often see only the top few, not all the recommendation list. Therefore, this proposed model is more suitable for the existing model.

Table 2. Shows the results of experiment

Results		
RMSE		
BW		Non-BW
0.36656		0.37894
Precision@K		
	BW	Non-BW
2	**0.87732**	0.86454
3	**0.76987**	0.76006
4	**0.67628**	0.67291
5	0.59407	**0.5948**
6	0.51439	**0.5236**
7	0.46276	**0.4663**
8	0.40517	**0.41216**
9	0.36241	**0.36925**
10	0.33278	**0.33427**

5 Conclusion

In this paper, we use K-Means and Matrix Factorization based ALS technique to develop a recommendation system that considers the clustering-based Bandwagon Effect by using in memory processing framework Spark. Consequently, the proposed model showed good accuracy for items with high ratings that can be practically recommended, and at the same time proved to be superior to previous models in terms of accuracy of recommending top content.

References

1. Kim, D., et al.: Research on cold-start recommendation. Commun. Korean Inst. Inf. Sci. Eng. **34**(6), 16–21 (2016)
2. Koren, Y., et al.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
3. Sundar, S., et al.: The Bandwagon Effect of collaborative filtering technology. In: Proceeding CHI EA 2008 on Human Factors in Computing Systems, pp. 3453–3458 (2008)
4. Choi, S. et al.: A recommendation model using the Bandwagon Effect for e-marketing purposes in IoT. Int. J. Distrib. Sens. Netw. **11**(7) (2015)
5. Kang, S., et al.: Improving diversity using Bandwagon Effect for developing recommendation system. Far East J. Elec. Commun. (2017, in Press)
6. GroupLens MovieLens Dataset (2016). http://grouplens.org/datasets/movielens/
7. Son, J., et al.: Review and analysis of recommender systems. J. Korean Inst. Ind. Eng. **41**(2), 185–208 (2015)