An Implicit Information Based Movie Recommendation Strategy

Jie Chen, Junjie Peng, Yingtao Wang, Gan Chen

School of Computer Engineering and Science, Shanghai University

Shanghai, China

ces_chenjie@shu.edu.cn, jjie.peng@i.shu.edu.cn, yingtaowang@i.shu.edu.cn, cgclio@shu.edu.cn

***Abstract*—Movie recommendation is a common way to attract audiences and enhance audience ratings in movie fields. The effectiveness of recommendation depends on the recommendation algorithm. To the traditional recommendation algorithms, such as user based collaborative filtering, matrix factorization based collaborative filtering and so on, the recommendation effect is strongly dependent on the users' ratings on the movies. However, in the actual movie online platform, the explicit rating data is very rare, which makes the recommendation result not very satisfactory. In view of this situation, this paper proposes an implicit information based recommendation strategy which can be used to establish user's preferences and make recommendation. By analyzing the watching logs, the strategy draws out the historical behavior and preference information of users which can be utilized as the basis of recommendation. Extensive experiments show that the strategy is correct, it can effectively analyze users' preferences on different factors on different movies. Meanwhile, it can make recommendation with high accuracy which can be used as the basis of recommendation without explicit information.**

Keywords—Movie Recommendation, Collaborative Filtering, Implicit Information, Recommendation Strategy

I. INTRODUCTION

The rapid development of the mobile Internet has injected new vitality into the development of the video industry, and the traditional video industry has shifted to the mobile online video industry. An audio-visual Development Research Report [1] points out that in 2017 the scale of internet video users in China reached 565 million. And 95% of video users chose watch the online video with mobile phones. There were 555 online dramas with 6921 episodes, 5620 online movies, 659 Internet animations and 2725 professional programs in the database of the State Administration of Radio Film and Television from January 1 to October 31, 2017. These are huge number of film and movie resources. And the number of these kinds of amusement movies increase more and more rapid with the development of internet technology. This has led to great burden for users to find the right movies that he is interested in. As he has to cost more time to choose and try before he find the appropriate one.

To make audiences easy to find suitable video programs, video recommendation is an effective and commonly used solution. Video recommendation system [2] is a kind of intelligent personalized information service system, which can describe the user's long-term information requirements with the help of user modeling technology based on a certain recommendation strategy. It is a common way to attract audiences and enhance audience ratings in online video fields. Its effectiveness depends on the recommendation algorithm. At present, video recommendation system mainly use the collaborative filtering recommendation method [3]-[5], the matrix factorization based recommendation method [6], etc. The effectiveness of these algorithms is strongly dependent on the user's explicit ratings for videos. However, in the actual video online platform, the explicit rating data is very rare. This makes the recommendation result not very satisfactory.

In view of this situation, this paper proposes an implicit information based recommendation strategy. That is, through analyzing the implicit information in users' behavior logs, we try to draw out the users' profiles [7] [8]. Usually profile includes the basic tags and attributes such as gender, age, education, location and so on, and preferences for different tags and attributes such as character, hobbies and interests. They can be used to analyze users' social information, behavior information and other data. It's promising and widely used in recommendation systems. In this paper, an implicit rating based is put forward to draw out the user profile based on which recommendation strategy is presented. We analyze the historical behavior and preference information of users by their online watching logs.

II. RELATED WORK

Recommendation has been widely used in many area. For example, Reference [9] proposed an all-weighted matrix factorization and fast optimization strategy for effective and efficient recommendation according to a frequency-aware weighting scheme. Reference [10] designed a personalized movie recommendation system based on collaborative filtering. Reference [11] proposed an improved content based collaborative filtering algorithm for movie recommendations, which found the similarities of the Genres and tags between the content of one movie and the other which are liked by the users. Reference [12] designed a location-based movie recommender system using collaborative filtering for enhancing the accuracy and the quality of recommendations rely on other useful available data such as users' locations. Reference [13] proposed a personalized travel route recommendation by estimating users' travel behavior frequencies by using collaborative filtering technique based on GPS trajectories.

Among recommendation methods, user profile based recommendation is one of the most popular methods that has been widely studied and applied in many fields. Reference [14] studied user profiling in the Twitter data of a large social media network, quantified the impact of a peer in the online environment of Twitter users and assessed the investment level of the identified social media users on a specific topic. In the study of drug vigilance, Reference [15] constructed user profiling based on the comments and related information of the drug, and combined the convolution neural network and the theme model to reliably predict the statistical attributes of the user such as age and gender. Reference [16] built user profiling for millions of researchers to dig their research interests using a technology called MagicFG based on web big data. Reference [17] studied the problem of dynamic user profiling in the context of streams of short texts using a streaming profiling algorithm. Reference [18] detected and classified antisocial activities (like trolling, spamming, fanatic posting, etc.) in commenting platforms by profiling, modeling and classifying on-line users' behavior.

User profiling concentrates on some attributes of users, so it has been widely studied and applied in the social network and commodity recommendation system. However in the field of movie the user profiling is more about the basic attributes of users and can't be used to make implicit rating on movies. And it does not consider that users' difference in implicit information such as preferences for different types of movies and so on. Therefore, implicit information based rating method and recommendation strategy in movie field have certain research value.

III. BASIC DEFINITION

To online movie platform, explicit comments or rating information about the movies from audiences may be limited. However the number of log files is enormous. These common log files contain many information about audiences that cannot be obtained in explicit manner. The information includes login logs, watching logs, time logs etc. User's behavior trace can be found in these log files according to the user's ID. By analyzing it, the users' implicit information can be dug out, so the users' implicit preferences for different factors can be described accurately. Because these implicit preferences is obtained by users' behavior record, the effectiveness of the implicit information based recommendation strategy is very outstanding.

The set of all movies is C, the size of C is N_1 (N_1 is a positive integer), of which the ith movie C_i is:

$$C_i = \left(L_{C_i}, D_{C_i}, A_{C_i}, K_{C_i}, B_{C_i}, E_{C_i}\right),\ 1 \le i \le N_1 \quad (1)$$

Where L_{C_i} is the length of the movie, D_{C_i} is the list of directors, A_{C_i} is the list of actors, K_{C_i} is the list of tags, B_{C_i} is the list of classifications, E_{C_i} is the list of sub-classifications.

The set of login logs for user u is F_u, the size of F_u is N_2 (N_2 is a positive integer), of which the jth login log F_{uj} is:

$$F_{uj} = \left(M_{F_{uj}}, G_{F_{uj}}, T_{F_{uj}}\right),\ 1 \le j \le N_2 \quad (2)$$

Where $M_{F_{uj}}$ is the device model used by the user, $G_{F_{uj}}$ is the location of the user, and $T_{F_{uj}}$ is the login time.

The set of watching logs for user u is P_u, the size of P_u is N_3 (N_3 is a positive integer), of which the kth watching log P_{uk} is:

$$P_{uk} = \left(T_{P_{uk}}, L_{P_{uk}}, C_{P_{uk}}\right),\ 1 \le k \le N_3 \quad (3)$$

Where $T_{P_{uk}}$ is the start time, $L_{P_{uk}}$ is the length of watching time, and $C_{P_{uk}}$ is the movie that the user has watched ($C_{P_{uk}} \in C$).

The set of log data for user u is $\{F_u, P_u\}$.

IV. AN IMPLICIT INFORMATION BASED RATING METHOD

Each log in $\{F_u, P_u\}$ for user u contains a login timestamp $T_{F_{uj}}$ or a watching timestamp $T_{P_{uk}}$, which represents the time with the corresponding behavior happened. All the logs about user u can be extracted from $\{F_u, P_u\}$ according to its ID number. Then the user's behavior trace can be obtained by sorting these logs according to their timestamp from small to large, which is recorded as H_u.

A. Basic implicit information

Users in different places have different preferences on different types of movies, and users in the same area may have some common interests. Users may watch different types of movies with different devices, and users who have the same model or brand of device may have some common interests. So it is an important implicit information for recommendation.

The total using times of the model M for the user u is J_{uM}:

$$J_{uM} = \left|\left\{M_{F_{uj}}\right\}\right|,\ 1 \le \mathrm{j} \le \mathrm{N}_2,\ M_{F_{uj}} = M \quad (4)$$

Where $M_{F_{uj}}$ is the model in each login log F_{uj} in F_u(the set of login logs for user u), N_2 is the size of F_u, and |.| is the size of a set.

The total using times of the place G for the user u is J_{uG}:

$$J_{uG} = \left|\left\{G_{F_{uj}}\right\}\right|,\ 1 \le \mathrm{j} \le \mathrm{N}_2,\ G_{F_{uj}} = G \quad (5)$$

Where $G_{F_{uj}}$ is the place in each login log F_{uj} in F_u(the set of login logs for user u), N_2 is the size of F_u, and |.| is the size of a set.

The total length of watching time on actors reflects users' preferences on actor lists of different movies as a whole. For example, the total length of watching time on actor Liying Zhao for users who are the fans of Liying Zhao is much longer than that who are not. The total length of watching time on actor a for user u is:

$$I_{ua} = \sum_{k=1}^{N_3} L_{P_{uk}},\ a \in A_{P_{uk}} \quad (6)$$

Where $L_{P_{uk}}$ is the length of watching time in each watching log P_{uk} in P_u(the set of watching logs for user u), N_3 is the size of P_u, $A_{P_{uk}}$ is the actor list of the movie in P_{uk}.

The total length of watching time of the tags for users reflects their preference on tag lists of different movies as a whole. For example, for users who mainly watched the war movies, the total length of watching time of the tag 'war' is much longer than the tag 'campus'. The total length of watching time of the tag m for user u is:

$$I_{um} = \sum_{k=1}^{N_3} L_{P_{uk}},\ m \in K_{P_{uk}} \quad (7)$$

Where $L_{P_{uk}}$ is the length of watching time in each watching log P_{uk} of P_u(the set of watching logs for user u), N_3 is the size of P_u, $K_{P_{uk}}$ is the tag list of the movie in P_{uk}.

B. Simple rating

Simple ratings on movies for the users can be calculated according to their log data, and it can be used as the input of the collaborative filtering algorithm.

The simple rating of movie C_i for user u is S_{uC_i}:

$$S_{uC_i} = \begin{cases} s, & 0 \le s < 1 \\ 1, & s \ge 1 \end{cases}, s = \sum_{k=1}^{N_3} \frac{L_{P_{uk}}}{L_{C_i}}, C_{P_{uk}} = C_i \quad (8)$$

Where N_1 is the size of the C (the set of all movies), N_3 is the size of P_u (the set of watching logs for user u), for each watching log P_{uk} in P_u, $L_{P_{uk}}$ is the length of watching time of P_{uk}, $C_{P_{uk}}$ is the movie in P_{uk}, L_{C_i} is the length of the movie C_i.

The simple rating reflects the user's finishing percentage of a movie. Considering that a user may not be able to finish watching a movie continuously, S_{uC_i} is defined as the sum of all finishing percentages of movie C_i for user u. For example, there is a movie with a length of 45 minutes, a user may have watched it for 20 minutes and 25 minutes respectively, so the simple rating equals to 100%. Because of the users' backtracking and repeated watching, the total length of watching time of the movies may exceed the length of them, so when the simple rating is greater than 100%, it is assigned to 100%.

The simple rating can solve the problem of rare explicit ratings that the traditional recommendation algorithms such as user based collaborative filtering and matrix factorization based collaborative filtering cannot solve. According to the matrix of simple ratings, similarities between users or movies can be calculated using a distance measure such as Cosine Similarity. For a user, movies that other users who are the most similar persons to him have watched but he hasn't will be recommended to him. And movies that he hasn't watched and are the most similar to what he has watched will be recommended to him.

C. *Total position weighted rating*

Each user has some simple ratings on movies he watched. The list of tags for each movie contains some tags which have different effects on profiling the movie. The list of actors and directors of each movie contains some actors and directors who play different roles in it and have different importance and contributions. The usual practice of movie industry is to place more important factors such as tags, actors, directors, classifications and so on in front of other factors. For each factor in the movie involved in a simple rating, the user's preference for it should be weighted according to its position in the list in which it is located. So on the basis of this simple rating, users' position weighted ratings on different factors in each movie can be constructed. Based on these position weighted ratings, the total position weighted ratings for all movies on these factors are constructed, which reflect their total preferences for these factors.

The total position weighted rating on factor p (a tag, an actor, etc.) for user u is W_{up}:

$$W_{up} = \sum_{i=1}^{N_1} S_{uC_i} * \alpha_p \quad (9)$$

Where N_1 is the size of C (the set of all movies), for each movie C_i, S_{uC_i} is the simple rating of C_i for u, α_{p} is a position weight of p in C_i. In this paper,

$$\alpha_p = \frac{1}{I_{pi}} \quad (10)$$

Where I_{pi} is a positive integer. It represents the position of p in the list in which it is located of the movie C_i, if p does not exist in any list of C_i, then I_{pi} gets $+\infty$ and α_{p} gets 0.

For example, the total position weighted rating on tag m for user u is W_{um}:

$$W_{um} = \sum_{i=1}^{N_1} S_{uC_i} * \alpha_m \quad (11)$$

Where N_1 is the size of C (the set of all movies), for each movie C_i, S_{uC_i} is the simple rating of C_i for u, α_{m} is a position weight of m in C_i. In this paper,

$$\alpha_m = \frac{1}{I_{mi}} \quad (12)$$

Where I_{mi} is a positive integer. It represents the position of m in the K_{C_i}(tag list of C_i), if m does not exist in K_{C_i}, then I_{mi} gets $+\infty$ and α_1 gets 0.

Similarly, the total position weighted rating on actor a for user u is W_{ua}, the total position weighted rating on director d for user u is W_{ud}, The total position weighted rating on classification b for user u is W_{ub}, the position weighted rating on sub-classification e for user u is W_{ue}.

D. *Normalized total position weighted rating*

These above total position weighted ratings on different factors for users reflect their total preferences for some factors (tags, stars, etc.). However, for the same type of these factors, different users can not directly compare ratings above with others because these ratings have not been normalized. For example, the total position weighted ratings for user u_1 and user u_2 on actor Liying Zhao are 0.4 and 0.3 respectively, and on actor Yuanyuan Gao are 0.8 and 0.1, and they have no other ratings on other actors, so, if we only compare their ratings above directly, we will conclude that user u_1 prefers Liying Zhao to user u_2 (0.4 > 0.3). However in fact, the normalized preference for u_1 to Liying Zhao is 0.4 / (0.4 + 0.8) = 0.33 while the normalized preference for u_2 to Liying Zhao is 0.3 / (0.3 + 0.1) = 0.75. It is reasonable that user u_2 prefers Liying Zhao to user u_1(0.33 < 0.75). Therefore, it is necessary to normalize the total position weighted ratings for each user. The normalized total position weighted ratings reflect the percentage of users' overall preference of different factors, so that normalized preferences for different users can be compared.

The normalized total position weighted rating on factor p (a tag, an actor, etc.) for user u is Q_{up}:

$$Q_{up} = \begin{cases} 0, & \sum_p^{\cup\{p\}} W_{\mathrm{up}} = 0 \\ \frac{W_{\mathrm{up}}}{\sum_p^{\cup\{p\}} W_{\mathrm{up}}}, & \sum_p^{\cup\{p\}} W_{\mathrm{up}} > 0 \end{cases} \quad (13)$$

Where $\cup\{p\}$ is the union of elements that of the same kind of p in all movies.

For example, the normalized total position weighted rating of tag m is Q_{um}:

$$Q_{um} = \begin{cases} 0, & \sum_m^K W_{\mathrm{um}} = 0 \\ \frac{W_{\mathrm{um}}}{\sum_m^K W_{\mathrm{um}}}, & \sum_m^K W_{\mathrm{um}} > 0 \end{cases} \quad (14)$$

$$K = K_{C_1} \cup K_{C_2} \cup \cdots K_{C_{N_1}} \quad (15)$$

Where K_{C_i} is the tag list of C_i, K is the union of all K_{C_i}, N_1 is the size of C (the set of all movies).

Similarly, the normalized total position weighted rating on actor a is Q_{ua}, the normalized total position weighted rating on director d is Q_{ud}, the normalized total position weighted rating on classification b is Q_{ub}, the normalized total position weighted rating on sub-classification e is Q_{ue}.

E. *Implicit rating*

The implicit rating on movie C_i for user u can be calculated by above normalized total position weighted ratings, it is made up of the following parts:

The total weighted rating on K_{C_i}(tag list of C_i) for user u:

$$Y_1 = \beta_1 * \sum_m^{K_{C_i}} Q_{um} * \alpha_m \quad (16)$$

Where β_1 is the weight of Y_1, the greater β_1, the more important the tag list is, α_{m} is a position weight for the tag m in C_i.

The total weighted rating on A_{C_i}(actor list of C_i) for user u:

$$Y_2 = \beta_2 * \sum_a^{A_{C_i}} Q_{ua} * \alpha_a \quad (17)$$

Where β_2 is the weight of Y_2, the greater β_2, the more important the actor list is, α_{a} is a position weight for the actor a in C_i.

The total weighted rating of D_{C_i}(director list of C_i) for user u:

$$Y_3 = \beta_3 * \sum_d^{D_{C_i}} Q_{ud} * \alpha_d \quad (18)$$

Where β_3 is the weight of Y_3, the greater β_3, the more important the director list is, α_{d} is a position weight for the director d in C_i.

The total weighted rating of B_{C_i}(classification list of C_i) for user u:

$$Y_4 = \beta_4 * \sum_b^{B_{C_i}} Q_{ub} * \alpha_b \quad (19)$$

Where β_4 is the weight of Y_4, the greater β_4, the more important the classification list is , α_{b} is a weight for the classification b in C_i.

The total weighted rating of E_{C_i}(sub-classification list of C_i) for user u:

$$Y_5 = \beta_5 * \sum_e^{E_{C_i}} Q_{ue} * \alpha_e \quad (20)$$

Where β_5 is the weight of Y_5, the greater β_5, the more important the sub-classification list is, α_e is a weight for the sub-classification e in C_i.

The implicit rating of the movie C_i for user u is Z_{uC_i}:

$$Z_{uC_i} = Y_1 + Y_2 + \cdots + Y_5 \quad (21)$$

V. AN IMPLICIT INFORMATION BASED RECOMMENDATION STRATEGY

Cold start is a widespread phenomenon that must be considered in the recommendation system. Cold start occurs when there is only a small number of logs for the user in the recommendation system. So it is difficult to calculate the user's implicit preference when it occurs. To judge it, we define a positive integer threshold N_7. The size of $\{F_u, P_u\}$ (the logs of user u) is N_6. There is a cold start when $N_6 < N_7$.

The implicit ratings reflects users' preferences for different factors (tags, actors, directors, etc.). For the movies that was not watched by a user, the implicit rating reflects the user's total preferences of those movies. So the higher the implicit rating of a movie is, the more worthwhile it is recommended to the user. Therefore, the recommendation strategy based on implicit rating is as follows:

Step 1: Judge whether there is a cold start. If "cold start" occurs, for the user, movies are put in the recommendation list that watched by other users who use the same model or brand of device with him or are in the location closed to him. This is because users' interests are often influenced by the people around them and users who use the same model or brand of device often share the similar interests. Then jump to the step 7. Otherwise, jump to the step 2.

Step 2: Get the movie list the user have watched. To each movie in the list, calculate simple rating on it.

Step 3: To each movie in the list of the step 2, calculate the position weighted rating for each factor (actor, tag, etc.) in the list in which it is located of the movie.

Step 4: To each factor in the step 3, calculate the total position weighted ratings on it. Then normalize these factors to get the normalized total position weighted ratings on them.

Step 5: To each movie in the list of the step 2, calculate the implicit rating on it. Put them in the preliminary recommended list in ascending order of the implicit ratings.

Step 6: Define a positive integer threshold N_8. Filter each movie in the list in step 5, remove it from the list if the implicit rating on it is smaller than N_8. If the filtered list is empty, movies that are hot on the platform are put in the recommendation list. Otherwise, jump to the step 7.

Step 7: Provide the recommendation list. Exit the recommendation algorithm.

VI. EXPERIMENT AND RESULTS ANALYSIS

Extensive experiments have been carried out using the three month logs of a large online movie platform to verify and test the effectiveness of the implicit information based rating method and recommendation strategy.

A. *Experiment environment*

These experiments use three PC machines and use HDFS to store data, use Hive for data cleaning, use HBase to realize fast query and Spark for distributed computing. The experimental software and hardware environment are shown in Table Ⅰ.

In these experiments:

$$\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 1 \quad (22)$$

TABLE I. EXPERIMENTAL SOFTWARE AND HARDWARE ENVIRONMENT

Host	CPU	RAM	Software
1	Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz	4G	hadoop-2.7.3 spark-2.0.2-bin-hadoop2.7 apache-hive-2.3.2 hbase-1.2.6
2	Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz	8G	hadoop-2.7.3 spark-2.0.2-bin-hadoop2.7 apache-hive-2.3.2 hbase-1.2.6
3	Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz	8G	hadoop-2.7.3 spark-2.0.2-bin-hadoop2.7 apache-hive-2.3.2 hbase-1.2.6

The evaluating indicator of the experiments is Mean Average Precision based on Precision and Average Precision.

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \qquad (23)$$

Where U is a set contains all users, $R(u)$ is the recommendation list for user u, $T(u)$ is the set of movies user u watched, |.| is the size of a set. Considering that the recommendation list is orderly, for a user, the AP (Average Precision) of his recommendation list is the average of each precision for each position in the list. The MAP (Mean Average Precision) is the mean of all users' AP:

$$\text{MAP} = \frac{\sum_{u \in U} AP(R(u), T(u))}{|U|} \qquad (24)$$

Where U is a set contains all users, $R(u)$ is the recommendation list for user u, $T(u)$ is the set of movies user u watched, |.| is the size of a set.

B. Experiment and result analysis

The log files of the data set are stored in the HBase database, and the ID of user is used as the line key and the log time is used as the column name to find a large number of log data by user quickly. The user behavior trace can be obtained by the processing according to the timestamp of these logs, and then the implicit information in which can be found and implicit preferences of users can be calculated.

TABLE II. A USER'S BEHAVIOR TRACE

Log Type	Time	Watching Time	Model Device	Location	Movie
login	10:01:41		Vivo X20A	Baoding, Hebei	
watching	10:03:21	26 s			I AM NEZHA
login	14:10:25		Vivo X20A	Baoding, Hebei	
watching	14:18:40	8 min 6 s			When We Are Young

For example, a user's behavior trace on October 2, 2017 is shown in table Ⅱ, which provides a basis for user profiling.

As is shown in Table Ⅲ the most frequently used device model for a user is "Vivo X20A".

TABLE III. THE MOST FREQUENTLY USED DEVICE MODEL FOR A USER

Device Model	Using Times
Vivo X20A	16
Vivo Y66	2

As is shown in Table Ⅳ, the most frequently used place for a user is "Baoding, Hebei".

TABLE IV. THE MOST FREQUENTLY USED LOCATION OF A USER

Device Model	Using Times
Baoding, Hebei	16
Tangshan, Hebei	2

As is shown in Table Ⅴ, the tags that a user watched the longest are "Feature" and "Action". These data reflect the overall interests of users about different tags.

TABLE V. THE TOTAL LENGTH OF WATCHING TIME OF DIFFERENT TAGS FOR A USER

Tag	Total Length of Watching Time
Feature	3 h 4 min 3 s
Action	2 h 55 min 30 s
Youth	15 min 26 s
Military	7 min 20 s
Cartoon	3 min 5 s
Comedy	2 min 39 s

More results about other implicit information are not described one by one. Experimental results show the method proposed is correct. It can effectively build users' preferences with the implicit information in the log data.

In order to compare MAP of user based collaborative filtering algorithm and the implicit information based recommendation strategy, the data set contains 3 months' logs is divided: the first 2 months' data are used as the training set, and the data of the later 1 month' are used as the test set. In this experiment, user based collaborative filtering algorithm uses the Mahout (an open source project)' implementation named GenericUserBasothRecommender, in which the Euclidean Distance is selected as the distance metric method, and the nearest N_8 users are designated as neighbors. N_8 is designated as 100 and 1000 respectively in the experiment, and the corresponding names of these models are UserBased100 and UserBased1000 respectively. The MAP values of the recommendation lists generated by the user based collaborative filtering algorithm and the implicit information based recommendation strategy are calculated separately under the different values of TopN (the size of recommendation list).

As is shown in Figure 1, with the increase of TopN, the MAP values of the user based collaborative filtering algorithm and the implicit information based recommendation strategy both show a downward trend. This is because the MAP value takes into account the positions of the recommended movies in the recommendation list. According to the calculation process of the MAP value, as TopN increases linearly, the number of hit movies is increasing, however they are at the back of the recommendation list, so the AP value is decreasing sharply and the MAP value is decreasing sharply too. Meanwhile, the MAP value of the implicit information based recommendation strategy is far higher than the user based collaborative filtering algorithm in the same TopN value. It verified that the implicit information based recommendation strategy has a higher accuracy, because the implicit information based recommendation strategy give full play to the users' implicit preference for different factors such as tags, actors, directors, classifications and so on and put forward a solution when "cold start" happened.

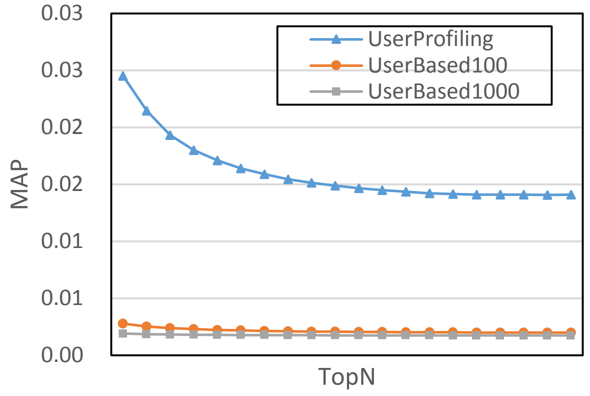

Figure 1. MAP of two recommendation methods with different TopN

VII. CONCLUSION

In view of the sparse or even lack of explicit ratings on movies for users in online video platform, an implicit information based recommendation strategy is proposed which provides a recommendation list according to the order of users' implicit ratings and put forward a solution when "cold start" happened.

Experiments show that the implicit information based recommendation strategy is more accurate than the traditional user based collaborative filtering algorithm. It gives full play to the users' preference for different factors such as different tags, actors, directors, classifications and so on, and has a certain application prospect.

REFERENCES

[1] China Netcasting Services Association, 2017 China Network audio-visual Development Research Report, 2017.

[2] L. Ren, Research on some Key Issues of Recommender Systems, East China Normal University, 2012.

[3] R. M. Bell, Y. Koren, Improved neighborhood-based collaborative filtering, KDD cup and workshop at the 13th ACM SIGKDD international conference on knowledge discovery and data mining, 2007.

[4] S. Ruslan, A. Mnih, G. Hinton, "Restricted Boltzmann machines for collaborative filtering", Proceedings of the 24th international conference on Machine learning, vol. 119, pp. 713-719, 2007.

[5] J. D. M. Rennie, N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction", Proceedings of the 22nd international conference on Machine learning, 2005.

[6] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model", Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426-434, 2008.

[7] P. Fei, Research on User Profiling Construction, Dalian University of Technology, 2017.

[8] H. Qiu, B. Shao, Construction of Academic Libraries' Accurate Service Based on User Portrait, Library Work in Colleges and Universities, vol. 38, no. 2, pp. 70-74, 2018.

[9] H. Li, X. Diao, J. Cao and Q. Zheng, Collaborative Filtering Recommendation Based on All-Weighted Matrix Factorization and Fast Optimization. IEEE Access, vol. 6, pp. 25248-25260, 2018.

[10] V. Subramaniyaswamy, L. Ravi, M. Chandrashekhar, A. Challa and V. Varadharajan, A personalised movie recommendation system based on collaborative filtering, IJHPCN, vol. 10, no. 1/2, pp. 54-63, 2017.

[11] A. Pal, P. Parhi, M. Aggarwal, An improved content based collaborative filtering algorithm for movie recommendations, IC3, pp. 1-3, 2017.

[12] K. Madadipouya, A Location-Based Movie Recommender System Using Collaborative Filtering, CoRR, vol. abs/1508.01696, 2015.

[13] G. Cui, J. Luo, X. Wang, Personalized travel route recommendation using collaborative filtering based on GPS trajectories, Int. J. Digital Earth, vol. 11, no. 3, pp. 284-307, 2018.

[14] M. Al-Qurishi, S. Alhuzami, M. A. AlRubaian , M. S. Hossain, A. Alamri, User profiling for big social media data using standing ovation model, Multimed Tools Appl, vol. 77, no. 9, pp. 11179-11201, 2018.

[15] E. Tutubalina, S. Nikolenko, Exploring convolutional neural networks and topic models for user profiling from drug reviews, Multimed Tools Appl, vol. 77, no. 4, pp. 4791-4809, 2018.

[16] X. Gu, H. Yang, J. Tang, Profiling Web users using big data, Soc. Netw. Anal. Min, vol. 8, no. 1, p. 24, 2018.

[17] S. LIANG, Dynamic User Profiling for Streams of Short Texts, AAAI Conference on Artificial Intelligence, North America, 2018.

[18] T. C. Li, "Profiling, Modeling and Classifying On-Line Users' Behavior", University of California, Riverside, USA 2017.