2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Analyzing Item Features for Cold-Start Problems in Recommendation Systems

Soryoung Kim, Sang-Min Choi, Yo-Sub Han
Department of Computer Science
Yonsei University
Seoul, Republic of Korea
{*soryoung, jerassi, emmous*}*@cs.yonsei.ac.kr*

Ka Lok Man, Kaiyu Wan
Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University
Suzhou, China
{*ka.man, kaiyu.wan*}*@xjtlu.edu.cn*

***Abstract*—With the development of web technologies, there is lots of information on the web. For effective web searching, the recommendation systems appear on the web. The recommendation systems provide customized information for the personal users. The conventional processes of the recommendation are generally based on the user preferences for the items. This leads to the cold–start problems for new items in recommending since new items have no user preferences. Although there are some studies to alleviate this problem by utilizing item features such as category information, the studies do not provide the validities of the use of item features. Namely, they just use the item features without analyzing features. If a feature draws meaningful recommendation results, there are some reasons that the feature can draw the results. We try to find these reasons. We calculate the uncertainty of item features by applying entropy in information theory and assume that this uncertainty of item features can show the level of reliability for the recommendation results. We verify our assumption by utilizing some tests in movie domain.**

***Keywords*-recommendation systems; feature uncertainty; feature entropy; cold-start problems;**

I. Introduction

Nowadays, with the increase of information on the web, users' choices are diversifying. When we search information on the web portal such as google, the portal provides lots of results for input keyword. In this situation, we cannot consider that all results are useful. Because of this reason, users have trouble in searching the information what they want. Thus, it is difficult works to provide information customized for each user on the web. Some researchers have studied recommendation systems to solve these kinds of problems [2], [9]. The recommendation systems are one of information filtering systems that predict users' preferences and provide users' preferred items based on their preference history [2]. Users can receive the information customized for their preferences through thus recommendation systems. For example, this system recommends an item in online shopping mall to a user based on this users purchase history data. One of real cases is book recommendation of the Amazon.com [7]. In addition, various web sites such as MovieLens[1] and Last FM[2] provide personalized recommendation services. Recently, the recommendation services extend their domain to not only commercial items but also recommendation of personal connection in social network service and personalized advertisements [4]. The conventional method for the recommendation systems is collaborative filtering [2], [9]. For example, in the movie recommendation services, the method recommends some movies to users based on results of analyzing users' preferences. If a user prefers Avengers, then the system recommends Ironman based on other users' preferences. Thus, this method first calculates the similarities between users or items and predicts preferences based on these similarities. One of most advantages of conventional filtering methods is that the system does not need complicated computation processes since the methods use only users' response histories for items. However, it is difficult to gather the users' preferences. According to Ricci et al. [8], users are reluctant to provide their preferences to systems. Because of this reason, there are some situations that the system cannot use users' preferences. The problems for these situations are sparsity and cold–start in the recommendation systems based on the collaborative filtering [6], [10]. We analyze item features excluding users' preferences for items to substitute preference information in recommendation systems. Namely, we propose a method that identifies item features to draw reliable recommendation results by utilizing a concept of entropy in information theory [11]. We can expect that our analyzing results support alleviating sparsity and cold–start problems in recommendation systems since we can use the features that draw reliable results than users' preferences. We first revisit the previous researches on utilizing item features and for the item–side cold–start problems in Section II. Next, we explain our approach that uses both the method of analyzing meaningful item features for recommendation using movie database in Section III. Then, we show the validity of our approach. Finally, we conclude this paper in Section V with future works.

II. Related Work

The item–side cold–start problems occur when new items are added in database. New items are excluded from the processes of recommendation and cannot be offered to users since these have no preferences by any users. Because of

[1] http://www.movielens.org
[2] http://www.last.fm

978-1-4799-5390-5/14 $31.00 © 2014 IEEE
DOI 10.1109/IIH-MSP.2014.48

IEEE computer society

this situation, some researchers have studied alleviating the item–side cold–start problems. One of famous approaches is the method using item features [1]. Generally, item has various features. For example, movie data have genre, director, actors, and nation as features. The known approach predicts the preferences of new items by analyzing item features [12], [5]. Furthermore, Choi et al.[4] have studied for predicting preferences of new items by using opinions of representative users extracted from user rating network and category feature.

III. Our Approach

We first explain the database that we used for our tests. We aggregate two open databases (GroupLens and HetRec) and crawl IMDB web pages to utilize various features for the movies. Then we show the method that check meaningful features.

A. Database

We join three different types of databases; GroupLens, HetRec, and IMDB. Table I shows each database. Advantages of IMDB are huge amount of items and various features than others. However, this dataset has no user information such as user preferences. We can gain user information through GroupLens database while this dataset does not provide various item features. HetRec database provides country information for each item. We aggregate these three datasets to analyze various features for movie recommendation. Table II shows join database. We use 71,567 users, 855,598 movies, and various movie features such as year, country, and genres. The number of users is same to the GroupLens database since only this dataset provides user preferences. Because of this reason, we use not the total number of IMDB movies but 855,598 movies since these items have users' preferences. In Table II, the features of the movies are five; year, country, genres, directors, and actors. In these features, year means the release year for a movie. We divide each release year as 10-year unit. For example, if there is a movie A released at 1954, this movie belong to 1950s movies. We have total nine years in join database (from 1930s to 2010s). We use genre classification of IMDB. In IMDB, there are 26 genres. Director, actor, and country mean the number of directors, actors, and countries for 855,598 movies in join database.

Table I
THREE MOVIE DATABASES

Database	Features (Size of each feature)
GroupLens	user (71,567), movie (10,681), rating (10,000,054)
HetRec	user (2,113), movie (855,598), country (72), genre (20), actor (95,321), director (4,060)
IMDB	movie (2,798,497), genre (26), year (9), actor (269,044), director (2,458,113)

Table II
JOIN DATABASE

Database	Features (Size of each feature)
Join DB	user (71,567), movie (855,598), year (9), country (72), genre (26), actor (269,044), director (2,458,113)

B. The method for searching meaningful features

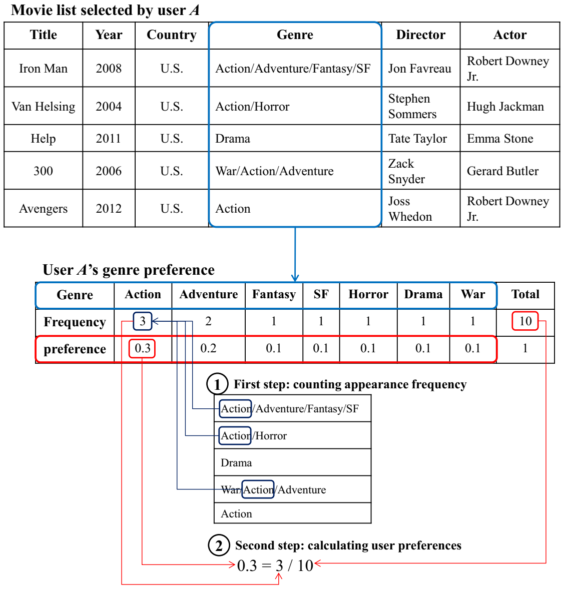
We first calculate user preferences for each feature to identify meaningful features. In this step, we do not use users' preferences. We draw the user preferences through other features for item based on information of user choice. Then we draw the feature uncertainty by utilizing the concept of entropy in information theory.

1) Calculating user preferences for each item feature: We use Equation (1) to calculate the user preferences for each feature.

$$UP_i = \frac{S_i}{\sum_{i=1}^{n} S_i}, \tag{1}$$

Figure 1 shows the example of calculating user preferences for a feature. There is user A who have selected total five movies and top table in Figure 1 shows these five movies with their features. Each movie has five features; year, country, genre, director, actor. In this example, we calculate the user preferences for genre. Namely, we show that calculate user preferences for genre as a feature. In user A's selected movies, seven genres appear. There are two steps to calculate the user preference for genre. First, we count appearance frequency of each genre. The second row on bottom table in Figure 1 shows frequencies for each genre. The action genre has three as appearance frequency since the action appears in Iron Man, Van Helsing, and 300. We repeat this process to all selected movies. Then we gain the appearance frequency for genre. After counting, we calculate the user preferences using Equation (1). The third row on bottom table in Figure 1 shows the results of Equation (1) for each genre. In this row, the action genre has 0.3 as preference since we divide frequency of the action genre, namely three, to total genre frequency 10. We can gain all genre preference by using these two steps.

2) Calculating feature uncertainty : We apply the entropy in information theory [11] to calculate uncertainty of each feature. Equation (2) shows the feature uncertainty. The results of this equation are same to the results of entropy calculation. Namely, we can gain the entropy of each feature through Equation (2). We define the entropy of a feature as the feature uncertainty. In information theory, we can check the uncertainty of random variable by utilizing the entropy. We consider a feature as a random variable and calculate the entropy to check uncertainty of each feature. Thus, the bigger result of Equation (2) is, the more uncertainty of the



Movie list selected by user *A*

Title	Year	Country	Genre	Director	Actor
Iron Man	2008	U.S.	Action/Adventure/Fantasy/SF	Jon Favreau	Robert Downey Jr.
Van Helsing	2004	U.S.	Action/Horror	Stephen Sommers	Hugh Jackman
Help	2011	U.S.	Drama	Tate Taylor	Emma Stone
300	2006	U.S.	War/Action/Adventure	Zack Snyder	Gerard Butler
Avengers	2012	U.S.	Action	Joss Whedon	Robert Downey Jr.

User *A*'s genre preference

Genre	Action	Adventure	Fantasy	SF	Horror	Drama	War	Total
Frequency	3	2	1	1	1	1	1	10
preference	0.3	0.2	0.1	0.1	0.1	0.1	0.1	1

1 **First step: counting appearance frequency**

- Action/Adventure/Fantasy/SF
- Action/Horror
- Drama
- War/Action/Adventure
- Action

2 **Second step: calculating user preferences**

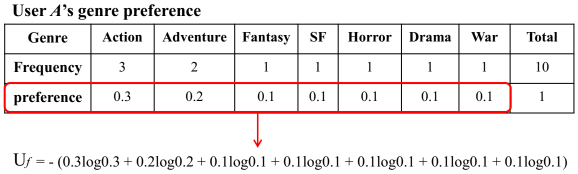
0.3 = 3 / 10

Figure 1. The example of calculating user preference for a feature

recommendation results exists on a feature.

$$U_f = -\sum_{i=1}^{n} UP_i logUP_i, \qquad (2)$$

In Equation (2), UP_i is the result of Equation (1) for i^{th} element in a feature. Figure 2 shows the example of computing feature uncertainty for genre by utilizing user preferences drawn from Figure 1. We first calculate the user preferences for each feature based on the number of choice by all users (71,567 users) in the database. Then we draw the feature uncertainty for each feature by utilizing the user preferences. Figure 3 shows the uncertainty for each feature in the database. In Figure 3, y–axis is feature uncertainty and x–axis is each feature. Genre has minimum feature uncertainty in this figure. It means that the uncertainty of genre is lower than other features and the results of recommendation in utilizing genre are more accurate than other features. On the other hand, the maximum value appears in actor. This implies that if we use the feature actor in recommending movies, the system draws adverse prediction results than in utilizing other features.



User *A*'s genre preference

Genre	Action	Adventure	Fantasy	SF	Horror	Drama	War	Total
Frequency	3	2	1	1	1	1	1	10
preference	0.3	0.2	0.1	0.1	0.1	0.1	0.1	1

U_f = - (0.3log0.3 + 0.2log0.2 + 0.1log0.1 + 0.1log0.1 + 0.1log0.1 + 0.1log0.1 + 0.1log0.1)

Figure 2. The example of calculating feature uncertainty

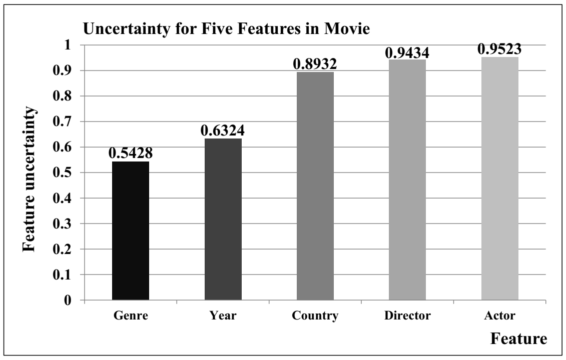


Figure 3. The graph for uncertainty of the five features

IV. Test and Analysis

We employ the recommendation systems based on user–based collaborative filtering [9], [13] to test validation of the feature uncertainty. In our system, we utilize cosine similarity to calculate the user similarities [9]. The conventional systems utilize user preferences, namely, users rating for movies, to draw the prediction scores [ref, ref]. In our test, we do not use rating information whereas our system utilizes the results of Equation (1) for five features as user preferences. We also address 10-fold cross validation [4] and mean absolute error (MAE) [3] for the tests. We divide all users by 10 equal folds. In the database, there are 71,567 users. Thus, each fold has approximately 7,156 users. One of 10 folds is selected as probe user set and other 9 folds are selected as training user set. Then we randomly select one movie. Then we extract the users who have selected this movie in the probe user set. After user extraction, we calculate the similarities between extracted users and the users in other 9 folds. Then we draw the prediction scores of the movie for all extracted users. Finally, we compute MAE for the prediction scores.

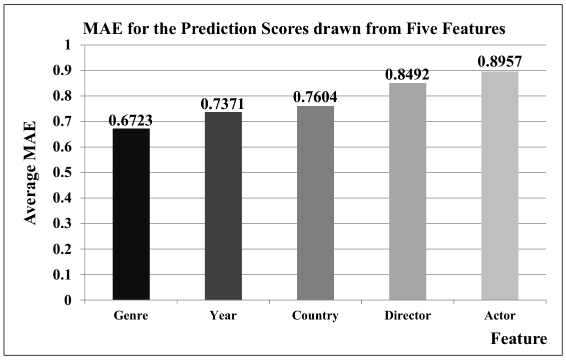


Figure 4. The graph for Average MAE of the five features

Figure 4 shows the average MAE for the prediction scores drawn from all folds. In Figure 4, y–axis and x–axis are average MAE and each feature respectively. We can see that the uncertainty of each feature and the average MAE are directly proportional to each other. Namely, the lower uncertainty guarantees the lower average MAE. It means that the recommendation systems utilizing a feature that have lower uncertainty than others can draw more precise prediction scores. Table III shows the values for average MAE and uncertainty for each feature simultaneously. Figure 4 and Figure III indicate that the accuracy of the recommendation results decreases according to increase of uncertainty for feature. Namely, we can infer the accuracy of recommendation results through feature uncertainty. Thus, we can provide meaningful item features for the precise recommendation, if there are no user preferences for items.

Table III
AVERAGE MAE AND UNCERTAINTY FOR EACH FEATURE

Feature	Feature uncertainty	Average MAE
$Genre$	0.5428	0.6723
$Year$	0.6324	0.7371
$Country$	0.8932	0.7604
$Director$	0.9434	0.8492
$Actor$	0.9523	0.8957

V. CONCLUSIONS

With the developments of web technologies, there is lots of information on the web. Because of this reason, the recommendation systems appear on the web. The systems effectively provide lots of information to users. However, the recommendation systems have one of critical problems called cold–start [10].

Although, the item features are generally utilized to alleviate the cold–start problems for new items [1], [12], [5], there are no validities for the use of each feature of items. We have analyzed item features to substitute preference information in recommendation systems. We have also proposed the method that identifies item features to draw reliable recommendation results by utilizing a concept of entropy in information theory [11]. We called the entropy of feature the uncertainty of feature. We have shown the uncertainty of each feature for movie domain by applying the entropy in information theory. We have assumed that this uncertainty of each feature can show the level of reliability for the recommendation results when we use each feature for recommendation than user preferences. Our test results have shown that our assumption is correct. Namely, the uncertainty drawn by our proposed methods can show the uncertainty of the recommendation results. It means that if there is a recommendation system that employs not user preferences but item features to recommend items, the system can grasp the each reliability of the recommendation results according to each feature through the uncertainty. If we apply this uncertainty to the methods for the cold–start problems of item–side, we can draw more precise recommendation results based on features that guarantee the reliabilities of the results.

ACKNOWLEDGEMENT: This research is partially supported by Grant from National Natural Science Foundation of China (Project Number 61103029).

REFERENCES

[1] S. S. Anand and N. Griffiths. A market-based approach to address the new item problem. In *Proceedings of the 2011 ACM Conference on Recommender Systems*, pages 205–212, 2011.

[2] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *The 15th International Conference on Machine Learning*, pages 46–54, 1998.

[3] M. Bulmer. *Principle of Statistics*. Dover Publications, 1979.

[4] S.-M. Choi and Y.-S. Han. Representative reviewers for internet social media. *Expert Systems with Applications*, 40(4):1274–1282, 2013.

[5] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 176–185, 2010.

[6] Z. Huang, H. Chen, and D. D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, 2004.

[7] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[8] F. Ricci and Q. N. Nguyen. Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent Systems*, 22(3):22–29, 2007.

[9] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference*, pages 285–295, 2001.

[10] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, 2002.

[11] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, Jan. 2001.

[12] D. Sun, Z. Luo, and F. Zhang. A novel approach for collaborative filtering to alleviate the new item cold-start problem. In *2011 International Symposium on Communications and Information Technologies*, pages 402–406, 2011.

[13] H. Xiong, J. Wu, and J. Chen. K-means clustering versus validation measures: A data distribution perspective. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 779–784, New York, NY, USA, 2006. ACM.