Improving the attribute-based active learning by clustering the new items

Junxin Zhou *†, Raja Chiky†
*Shenzhen University, China
Email: junxin.zhou@isep.fr
†ISEP, LISITE Lab, Paris, France
Email: raja.chiky@isep.fr

***Abstract*—The issue that recommender system often meets is cold-start problem, where the system does not have any ratings of new items or new users. Thus, it can not provide relevant recommendation for the new users or new items. In previous research, when dealing with item cold-start problem, some scientists combined content information and active learning method, and used factorization machine to model the prediction task. However, a shortcoming in this method is that when using factorization machine model to select users to give ratings to new items, the active users may be selected for too many times, leading to a result that they refuse to give ratings for new items, or randomly give their ratings, which does not exactly show their preferences. In this paper, to solve this issue, we use clustering algorithm to divide new items into different groups and choose one item to represent the group, and only request users giving ratings for the representative items.**

***Index Terms*—Active learning, Clustering**

I. Introduction

Recommendation systems aim to predict users preferences, which will result in better product sales and more revenue for companies. The most common method applied in recommender system is collaborative filtering. It suggests to users the most relevant items based on predefined features and past user activities. However, when it comes to cold-start problems, the recommended systems do not perform well. Cold-start problems include user cold-start problems and item cold-start problems. The user cold-start problem refers to joining of new users in the system; the item cold-start problem refers to push of new items. In this paper, we focus on the new item cold-start problem, where the system tries to recommend new items that dont have any ratings to users who really like them.

A common approach to mitigate the item cold-start problem is to exploit content information, such as item attributes. [1] combines exploiting the attributes of new items and active learning approach, which tries to select some users to give their ratings for new items. Moreover, to get a better feedback in active learning strategy, it applies Factorization Matrix to model the task for selecting users to rate new items.

However, in reality, when there are some users who are selected so many times to give ratings for new items, they may be sick of answering how they think of the new items. Under this circumstance, they might refuse to give their ratings or give ratings that dont exactly show their preferences (i.e. when they dont want to answer, they might give the same rating for every new item). In this paper, to mitigate that issue, we divide the new items into different groups according to their attributes. For the clustering method, we choose K-Means algorithm.

Thus, in this paper, with new items attribute and users feedback in active learning, we firstly apply K-Means algorithm to divide new items into different clusters, and then use factorization machine model to predict if users will give their ratings for new items and predict users preferences on new items.

II. Related work

A. Item-cold start problem

Roughly speaking, item cold-start problem is referred to systems incapability of dealing with new items due to the lack of relevant ratings. A common solution, which is known as hybrid approach, to solve this issue is to combine content information and collaborative filtering [2], [3] In [2] authors perform hybrid recommendation based on Boltzmann machines. [3] exploit the manifold structure and also combines content information and collaborative filtering to solve the item cold-start problem. Another approach is to deal with the situation that the system does not have any items attributes. [4], [5]use a linear model to estimate new items latent factors through users latent factors, which can derive from users previous ratings.

B. Active Learning

Active Learning is mostly used to deal with user cold-start problem in previous research. Generally, the strategy is to ask the new user to rate a certain number of items, and the feedback will eventually help the system to predict the new users preference. However, whether the items that are selected to be rated by new users are informative or not will have a significant effect on final result. In [6] it compares 10 non-personalized active leaning strategies in collaborative filtering, and finally there are different results between them. Similarly, when it comes to item cold-start problem, it is also important to select the relevant users to rate the new items. [1] try to find out the users that are more likely to be willing to rate new items and generate objective ratings. In our paper, for selecting users to rate new items and predicting users preferences on new items, we mainly base our research on the work from

978-1-7281-3851-0/19/$31.00 ©2019 IEEE
DOI 10.1109/SERVICES.2019.00095

[1] and improve it by adding a clustering phase to avoid soliciting the same user several times. [6] summarizes the clustering algorithms that can be applied to recommender system, consisting of K-means algorithm, density-based clustering, message-passing clustering and hierarchical clustering. In this paper, the clustering method we choose is K-means algorithm since it is the most common one used in recommender system and it has great simplicity and efficiency compared with the others

III. Our Approach

We use the same notation as in [1]. Let U, I and A be respectively the users, items and attributes. The objective is to predict the rating a chosen user u would give to a new item i_{new}.
$R \in R^{|U|\times|I|}$ is the user-item matrix, $T \in R^{|I|\times|A|}$ is a binary item-attribute matrix ($T[i,j]=1$ means that the item$_i$ contains the attribute A$_j$. Let's consider a matrix of new items for which the system has no or few ratings. We propose to solve the problem by following three steps:

1) Clustering step: We exploit new items attributes for the clustering task. After transforming the attributes into vectors, we start to apply K-means algorithm. However, there are two main issues. The first one is how we choose the initial clusters. This is important for K-means algorithm, because it can have a significant effect on the final result. We propose to use K-Means++ algorithm [7], which helps selecting the centers that almost have the largest distance with each other. Second, we have to decide the number of clusters carefully. To get a better result, we carry out the experimentations several times with different k (number of clusters). Then, for every cluster, we select the item that has the minimum distance to the center to represent the cluster.
2) Select users to rate the representative items: we use the factorization machine to model whether users will rate the new items. For information about this approach, we recommend the reading of [8]. Let $A_1..A_k$ the k representative item of the k clusters $(1..k)$. For every rating in previous dataset, we extract the users and the k representative items attributes and train them with the factorization machine. To represent the users, we use one-hot representation. $1,0,0,0$ is the first user U_1, $0,1,0,0$ is the second user, and so one. An example is given below The output $w(m)$ from the factorization machine means whether the user m will be willing to rate the new items, and the closer it is to 1, the more likely this user will rate the new item. Then, we calculate the variance of every users ratings, the smaller the variance is, the more objective this users ratings are. We take the variance of user m as $v(m)$. Finally, we combine the two factores $w(m)$ and $v(m)$ to get a final score t by using the formula: $t(m)=\frac{w(m)}{\lambda\times v(m)}$ where λ is a parameter that gives different weight to the factors. For every new item, we select the user having the largest $t(m)$ to rate it.
3) Predict users' ratings for the new items: For predicting the final result, we still use the factorization machine to train the previous ratings and the feedback we get from the active learning. Features in this model contain not only users and items attributes, but also items. The instances contain both previous ratings and the newly obtained feedback ratings.

IV. Evaluation

The implementation is ongoing when this paper is written. We plan to test our approach against the one presented in [1] using the same dataset movielens-Imdb. We will compare both approaches using the following metrics:

- PFR: Percentage of feedback rating. The higher PFR is, the better the result is.
$$PFR = \frac{total\ number\ of\ feedback\ ratings}{total\ number\ of\ rating\ requests}$$
- AST: Average Selecting Times. The lower AST is, the better the result is.
$$AST = \frac{total\ number\ of\ requests}{total\ number\ of\ distinct\ users\ for\ rating}$$
- RMSE: Root Mean Square Error. It measures the square root of the average of squared differences between prediction and actual observation. The lower the RMSE is, the better the result is.

A. Conclusion

As previously discussed, the active learning method may not have a good performance when the system requests the same user giving ratings for too many times. In this paper, we focus on item-cold start problem and to solve that issue, we apply a common clustering method K-means algorithm to divide the new items into different clusters. Moreover, we combine items attribute and collaborative filtering, and use matrix factorization to model the prediction task. Our ongoing work is to implement our proposed approach and compare it against the initial approach proposed by [1] to assess it efficiency.

References

[1] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, "Addressing the item cold-start problem by attribute-driven active learning," *CoRR*, 2018.

[2] A. Gunawardana and C. Meek, "Tied boltzmann machines for cold start recommendations," ser. RecSys '08, 2008.

[3] M. Saveski and A. Mantrach, "Item cold-start recommendations: Learning local collective embeddings," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14, 2014.

[4] M. Aharon, A. Kagian, R. Lempel, and Y. Koren, "Dynamic personalized recommendation of comment-eliciting stories," ser. RecSys '12, 2012.

[5] N. Aizenberg, Y. Koren, and O. Somekh, "Build your own music recommender by modeling internet radio streams," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12, 2012.

[6] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, in *Recommender Systems Handbook*, 2011.

[7] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[8] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, 2012.