Clustering-Based Collaborative Filtering Using an Incentivized/Penalized User Model

Kaneez Fatima [1], Amer Noor Khan [2], Saleha Butool [3]
[1, 2, 3] Assistant Professor,
Department of Information Technology,
LORDS Institute of Engineering & Technology, Hyderabad.

Abstract - Giving or recommending appropriate content based on the quality of experience is the most important and challenging issue in recommender systems. As collaborative filtering (CF) is one of the most prominent and popular techniques used for recommender systems, we propose a new *clustering- based* CF (CBCF) method using an incentivized/penalized user (IPU) model only with the ratings given by users, which is thus easy to implement. We aim to design such a simple clustering-based approach with no further prior information while improving the recommendation accuracy. To be precise, the purpose of CBCF with the IPU model is to improve recommendation performance such as *precision*, *recall*, and *F*1 score by carefully exploiting different preferences among users. Specifically, we formulate a constrained optimization problem in which we aim to maximize the *recall* (or equivalently *F*1 score) for a given *precision*. To this end, users are divided into several clusters based on the actual rating data and Pearson correlation coefficient. Afterward, we give each item an *incentive/penalty* according to the preference tendency by users within the same cluster. Our experimental results show a significant performance improvement over the baseline CF scheme without clustering in terms of *recall* or *F*1 score for a given *precision*.

Keywords: Clustering, collaborative filtering, *F*1 score, incentivized/penalized user model, Pearson correlation coefficient, recommender system.

Introduction

People are likely to have an increasing difficulty in finding their favorite content effectively since extensive collections of video, audio, papers, art, etc. have been created both online and offline. For example, over hundreds of feature films and hundreds of thousands of books have been produced and published every year in the US. However, one person would read at most about 10,000 books in his/her life, and then he/she must choose his/her favorite books among them. On the one hand, recommender systems have been developed and used in diverse domains (e.g., the movie industry, the music industry, and so on) by helping people to select appropriate content based on individual preferences [1].

Especially, online commerce industries such as Amazon.com and Netflix have successfully

exploited how to increase customer loyalty. For example, Amazon.com and Netflix have generated much of their sales by providing personalized items through their own recommender systems [2], [3].

While diverse recommender systems such as personalized recommendations, content-based recommendations, and knowledge-based recommendations have been developed, collaborative filtering (CF) is one of the most prominent and popular techniques used for recommender systems [4], [5].

CF methods are generally classified into memory-based CF and model-based CF. In model-based CF, training datasets are used to develop a model for predicting user preferences. Different machine learning techniques such as Bayesian networks, clustering, and rule-based approaches can also be utilized to build models. An alternating least squares with weighted regularization (ALS-WR) scheme is a representative example of model-based CF. ALS-WR is performed based on a matrix factorization algorithm and is tolerant of the data sparsity and scalability [6], [7]. The main advantages of model-based CF are an improvement of prediction performance and the robustness against the data sparsity.

However, it has some shortcomings such as an expensive cost for building a model [5]. On the other hand, memory-based CF does not build a specific model, but directly computes the similarity between users or items using the entire rating matrix or its samples. Hence, memory-based CF is easy to implement and effective to manage. However, it has also some drawbacks such as dependence on human ratings, performance decrement when data are sparse, and disability of recommendation for new users (i.e., cold-start users) and items [5].

Memory-based CF approaches are again classified into user-based CF and item-based CF. The main ideas behind the user-based CF and item-based CF approaches are to find the user similarity and the item similarity, respectively, according to the ratings (or preferences). After finding similar users, called *neighbors*, user-based CF recommends the top-N most preferable items that an active user has not accessed yet. User-based CF has limitations related to scalability, especially when the number of users is much larger than the number of items. Item-based CF was proposed to mitigate this scalability problem, but cannot still entirely solve the problem when the numbers of users and items are large. Despite such limitations, CF has been employed as one of the most representative recommender systems leveraged in online commerce.

In addition, there have been many studies on the design of CF algorithms in terms of reducing the mean absolute error (MAE) or root mean squared error (RMSE) of rating prediction [8]. However, recommender systems designed in the sense of minimizing the MAE or RMSE do not inherently improve recommendation accuracy. We assume that there are two recommender systems having the same MAE or RMSE of the rating prediction. We note that they may differ from each other in terms of user experience (UX) since there is a possibility that one recommender system recommends an item whereas the other does not.

For example, suppose that the real preference of a user on an item is 4.2 and two recommender systems predict the preference as 3.8 and 4.6, respectively. Then, when items having the predicted preference of more than 4.0 are assumed to be recommended, the MAEs of two recommender systems are the same but only the latter one will recommend the item. In order to redeem the above case, some performance metrics related to UX such as *precision*, *recall*, and $F1$ score have been widely used in the literature.

On the other hand, several companies, e.g., Pandora Internet Radio, Netflix, and Artsy, have developed their own clustering-based recommendation methods, called Music Genome Project, Micro-Genres of Movies, and Art Genome Project, respectively. These clustering-based recommendation methods have successfully led to satisfactory performance, but the processing cost for clustering is very expensive. For example, it is widely known that each song tends to be analyzed by a musician through a process that

takes usually 20 to 30 minutes per song in the case of Music Genome Project.

Related Work

G. Adomavicius and A. Tuzhilin, This paper presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multicriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

G. Linden, B. Smith, and J. York, Recommendation algorithms are best known for their use on e-commerce Web sites, where they use input about a customer's interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favorite artists. At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother. There are three common approaches to solving the recommendation problem: traditional collaborative filtering, cluster models, and search-based methods. Here, we compare these methods with our algorithm, which we call item-to-item collaborative filtering. Unlike traditional collaborative filtering, our algorithm's online computation scales independently of the number of customers and number of items in the product catalog. Our algorithm produces recommendations in real-time, scales to massive data sets, and generates high quality recommendations.

Y. Koren, R. Bell, and C. Volinsky, As the Netflix Prize competition has demonstrated, matrix factorization models are superior to classic nearest neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels.

X. Su and T. M. Khoshgoftaar, As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. In this paper, we first introduce CF tasks and their main challenges, such as data sparsity, scalability, synonymy, gray sheep, shilling attacks, privacy protection, etc., and their possible solutions. We then present three main categories of CF techniques: memory-based, model-based, and hybrid CF algorithms (that combine CF with other recommendation techniques), with examples for representative algorithms of each category, and analysis of their predictive performance and their ability to address the challenges. From basic techniques to the state-of-the-art, we attempt to present a comprehensive survey for CF techniques, which can be served as a roadmap for research and practice in this area.

Y. Cai, H.-F. Leung, Q. Li, H. Min, CF is an important and popular technology for recommender systems. However, current CF

methods suffer from such problems as data sparsity, recommendation inaccuracy, and big-error in predictions. In this paper, we borrow ideas of object typicality from cognitive psychology and propose a novel typicality-based collaborative filtering recommendation method named TyCo. A distinct feature of typicality-based CF is that it finds "neighbors" of users based on user typicality degrees in user groups (instead of the corated items of users, or common users of items, as in traditional CF). To the best of our knowledge, there has been no prior work on investigating CF recommendation by combining object typicality. TyCo outperforms many CF recommendation methods on recommendation accuracy (in terms of MAE) with an improvement of at least 6.35 percent in Movielens data set, especially with sparse training data (9.89 percent improvement on MAE) and has lower time cost than other CF methods. Further, it can obtain more accurate predictions with less number of big-error predictions.

Y. Koren, and C. Volinsky, A common task of recommender systems is to improve customer experience through personalized recommendations based on prior implicit feedback. These systems passively track different sorts of user behavior, such as purchase history, watching habits and browsing activity, in order to model user preferences. Unlike the much more extensively researched explicit feedback, we do not have any direct input from the users regarding their preferences. In particular, we lack substantial evidence on which products consumer dislike. In this work we identify unique properties of implicit feedback datasets. We propose treating the data as indication of positive and negative preference associated with vastly varying confidence levels. This leads to a factor model which is especially tailored for implicit feedback recommenders. We also suggest a scalable optimization procedure, which scales linearly with the data size. The algorithm is used successfully within a recommender system for television shows. It compares favorably with well tuned implementations of other known methods. In addition, we offer a novel way to give explanations to recommendations given by this factor model.

Problem Formulation

The contribution of our work is to make a proper decision with which items should be recommended or not under the same MAE or RMSE in terms of improving UX (i.e., *recall* (or equivalently *F*1 score) for a given *precision*). For example, suppose that there are two items with the same predicted preference value given by 3.9. If a recommender system only suggests items whose predicted preference is over 4.0, then above two items will be dropped by the system. However, there may be some users who are satis_ed with the items, and thus UX will decrease in this case. In order to enhance the UX, we give each item an *incentive* or *penalty* according to the preference tendency by users. To this end, we *cluster* users into some groups and make a decision on which items are given the *incentive/penalty* based on a group that users belong to.

Proposed Implementation System

Unlike the aforementioned clustering-based recommendation methods that take long processing time to recommend items, we aim to design a simple but novel *clustering-based* CF (CBCF) method only with ratings given by users, which is thus easy to implement. That is, we design such a simple clustering-based approach with no further prior information while improving the recommendation accuracy.

To this end, in this paper, we introduce the CBCF method using an incentivized/penalized user (IPU) model in improving the performance of recommender systems in terms of *precision*, *recall*, and *F*1 score. More specifically, we present the CBCF method by carefully

exploiting different preferences among users along with clustering. Our proposed method is built upon a predicted rating matrix-based clustering that can drastically reduce the processing overhead of clustering. In our CBCF method, we aim to select items to be recommended for users along with clustering. To this end, users are divided into several clusters based on the actual rating data and Pearson correlation coefficient. Then, items are regarded as more important or less important depending on the clusters that the users belong to. Afterwards, we give each item an *incentive*/*penalty* according to the preference tendency by users within the same cluster. The main contributions of our work are summarized as follows.

- An easy-to-implement CBCF method using the IPU model is proposed to further enhance the performance related to UX.
- To design our CBCF method, we first formulate a con- strained optimization problem, in which we aim to maximize the *recall* (or equivalently $F1$ score) for a given *precision*.
- We numerically find the amount of incentive/penalty that is to be given to each item according to the preference tendency by users within the same cluster.
- We evaluate the performance of the proposed method via extensive experiments and demonstrate that $F1$ score of the CBCF method using the IPU model is improved compared with the baseline CF method without clustering, while *recall* for given (fixed) *precision* can be significantly improved by up to about 50%.

Clustering-based Recommender Systems

There has been diverse research to enhance recommendation accuracy by means of clustering methods. In CF and content-based filtering methods were conducted by finding similar users and items, respectively, via clustering, and then personalized recommendation to the target user was made. As a result, improved performance on the *precision*, *recall*, and $F1$ score was shown. Similarly communities (or groups) were discovered before the application of matrix factorization to each community.

In social activeness and dynamic interest features were exploited to find similar communities by item grouping, where items are clustered into several groups using cosine similarity. As a result of grouping, the K most similar users based on the similarity measure were selected for recommendation. The performance of user-based CF with several clustering algorithms including K-Means, self-organizing maps (SOM), and *fuzzy C-Means (FCM)* clustering methods. It was shown that user-based CF based on the FCM has the best performance in comparison with K-Means and SOM clustering methods. Moreover, several clustering approaches were studied in CF-aided recommender systems: heterogeneous evolutionary clustering was presented by dividing individuals with similar state values into the same cluster according to stable states; another dynamic evolutionary clustering by computing user attribute distances; and more recently, dynamic evolutionary clustering based on time weight and latent attributes was proposed.

Proposed Mechanism

The CBCF method recommends desirable items according to the result of item clustering and the preference tendency of each user using our IPU model.

The main contribution of our CBCF method using the IPU model is to give either an incentive or a penalty to each item based on N *Cic* (the average preference on item *i* of users

within cluster *Cc*), which depends on the result of clustering. As mentioned before, since there are empty elements in the rating matrix **R***CBCF* that users have not rated or accessed yet, the Euclidian distance between user vectors (i.e., row vectors in **R***CBCF*) cannot be accurately calculated. Hence, we use the Pearson correlation coefficient (PCC) in our work. PCC computes the correlation between two users' common ratings to measure their similarity, and thus needs two common ratings at least.

Dataset Collection

We use the MovieLens 100K dataset6 with the following attributes:

- 100K dataset have 100,000 anonymous ratings
- Ratings are made on a 5-star scale
- There are 943 users in 100K dataset
- Each user has at least 20 ratings.

Note that the sparsity (i.e., the ratio of the number of missing cells in a rating matrix to the total number of cells) of the rating matrix obtained from the MovieLens 100K dataset is 93.7%, which is high and often causes performance degradation. One popular solution to the data sparsity problem is the use of data imputation, which includes the zero injection method which zeros are given to some missing cells in a rating matrix and two matrix factorization-based methods that assign zeros or twos to all missing cells in a rating matrix. Even if such data imputation techniques are known to significantly improve the prediction accuracy,

Experimental Results

In this section, we evaluate the performance of our proposed CBCF method using the IPU model in terms of *precision*, *recall*, and *F*1 score. In our experiments, unless otherwise stated, item-based CF is adopted in our proposed method since it shows better performance on the accuracy of recommendation for memory-based CF, which will be verified later in this section. We use Apache Mahout8 whose goal is to build an environment for performing downstream machine learning tasks such as CF, clustering, and classification. It is assumed that the recommendation result is true when the following conditions are met:

- The real rating of an item recommended to a user is 4.0 or 5.0.
- The real rating of an item not recommended to a user is less than 4.0.

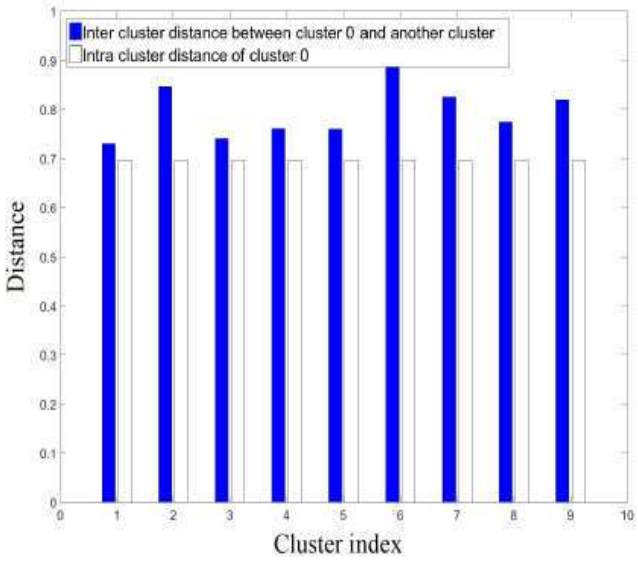

Figure1: Comparison of the inter-cluster and intra-cluster Euclidean distances.

In our experiments, the number of clusters for both *spectral* and FCM clustering algorithms is set to $c = 10$; the fuzzy degree m of FCM clustering is set to 2; and the convergence threshold of FCM clustering is set to 10^{-4}. In the FCM clustering, an object is assigned to such a cluster that has the highest coefficient. In our subsequent experiments, we adopt *spectral* clustering by default unless otherwise stated. Fig. 1 compares the inter-cluster Euclidean distances with the intra-cluster Euclidean distances in order to show the validity of clustering. The values of PCC range between -1.0 and 1.0, where 1.0 and -1:0 imply

that two objects (e.g., users) have the highest positive and negative correlations, respectively. Hence, since most clustering algorithms do not employ any negative correlation, the value of PCC between two users $u1$ and $u2$, namely $s(u1; u2)$, is shifted.

Generally, a small recommendation threshold value leads a low *precision* and high *recall*, and vice versa. However, as mentioned before, as the threshold value becomes very large, the $F1$ score is rapidly decreased because the decreasing rate of *recall* is faster than the increasing rate of *precision*.

Instead of item-based CF, user-based CF can also be employed in our proposed CBCF method. We evaluate the performance of our proposed CBCF method using the IPU model based on item-based CF and user-based CF methods, respectively where both *spectral* and FCM clustering algorithms are employed for non-cold-start users. Based on the results, the following observations are made:

i) The proposed method based on item-based CF achieves better performance on the $F1$ score than the case of user-based CF
ii) Using the proposed method based on FCM clustering is slightly superior to the case of *spectral* clustering.

Moreover, we evaluate the performance of the proposed and baseline methods for more difficult situations having cold-start users whose number of rated items is less than 20, where item-based CF and *spectral* clustering are used.

Conclusion

In this paper, we proposed a CBCF method using the IPU model in recommender systems by carefully exploiting different preferences among users along with clustering. Specifically, in the proposed CBCF method, we formulated a constrained optimization problem in terms of maximizing the *recall* (or equivalently $F1$ score) for a given *precision*. To this end, clustering was applied so that not only users are divided into several clusters based on the actual rating data and Pearson correlation coefficient but also an incentive/penalty is given to each item according to the preference tendency by users within a same cluster. As a main result, it was demonstrated that the proposed CBCF method using the IPU model brings a remarkable gain in terms of *recall* or $F1$ score for a given *precision*.

A possible direction of future research in this area includes the design of a new clustering-based CF method by exploiting the properties of model-based CF approaches (e.g., matrix factorization).

References

[1] G. Adomavicius and A. Tuzhilin, ``Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734_749, Jun. 2005.
[2] G. Linden, B. Smith, and J. York, ``Amazon.com recommendations: Item- to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76_80, Jan./Feb. 2003.
[3] Y. Koren, R. Bell, and C. Volinsky, ``Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30_37, Aug. 2009.
[4] J. A. Konstan and J. Riedl, ``Recommender systems: From algorithms to user experience," *User Model. User-Adapted Interact.*, vol. 22, nos. 1_2, pp. 101_123, Apr. 2012.
[5] X. Su and T. M. Khoshgoftaar, ``A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1_19, Jan. 2009, Art. no. 421425.
[6] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, ``Large-scale parallel collaborative filtering for the Netflix prize," in *Proc. Int.*

Conf. Algorithmic Appl. Manage., Shanghai, China, Jun. 2008, pp. 337_348.
[7] Y. Hu, Y. Koren, and C. Volinsky, ``Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008, pp. 263_272.
[8] Y. Cai, H.-F. Leung, Q. Li, H. Min, J. Tang, and J. Li, ``Typicality-based collaborative filtering recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 766_779, Mar. 2014.
[9] G. Guo, J. Zhang, and D. Thalmann, ``Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowl.-Based Syst.*, vol. 57, pp. 57_68, Feb. 2014.
[10] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, ``A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.- Based Syst.*, vol. 26, pp. 225_238, Feb. 2012.
[11] H. Sobhanam and A. K. Mariappan, ``A hybrid approach to solve cold start problem in recommender systems using association rules and clustering technique," *Int. J. Comput. Appl.*, vol. 74, no. 4, pp. 17_23, Jul. 2013.
[12] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, ``A new user similarity model to improve the accuracy of collaborative filtering," *Knowl.-Based Syst.*, vol. 56, pp. 156_166, Jan. 2014.
[13] B.-H. Huang and B.-R. Dai, ``A weighted distance similarity model to improve the accuracy of collaborative recommender system," in *Proc. 16th IEEE Int. Conf. Mobile Data Manage.*, Pittsburgh, PA, USA, Jun. 2015, pp. 104_109.
[14] Q. Lu, T. Chen,W. Zhang, D.Yang, andY.Yu, ``Serendipitous personalized ranking for top N recommendation," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Macau, China, Dec. 2012, pp. 258_265.
[15] K. Oku and F. Hattori, ``Fusion-based recommender system for improving serendipity," in *Proc. ACM Workshop Novelty Diversity Rec. Syst. (DiveRS)*, Chicago, IL, USA, Oct. 2011, pp. 19_26.
[16] P. Adamopoulos and A. Tuzhilin, ``On unexpectedness in recommender systems: Or how to expect the unexpected," in *Proc. ACM Workshop Novelty Diversity Rec. Syst. (DiveRS)*, Chicago, IL, USA, Oct. 2011, pp. 11_18.
[17] H. Andrat and N. Ansari, ``Analyzing game stickiness using clustering techniques," *Adv. Comput. Comput. Sci.*, vol. 554, pp. 645_654, Sep. 2017.
[18] K. Chowdhury, D. Chaudhuri, and A. K. Pal, ``A novel objective function based clustering with optimal number of clusters," *Methodologies Application Issues of Contemporary Computing Framework*, pp. 23_32, Sep. 2018.
[19] S.-H. Lee, Y.-S. Jeong, J.-Y. Kim, and M. K. Jeong, ``A new clustering validity index for arbitrary shape of clusters," *Pattern Recognit. Lett.*, vol. 112, pp. 263_269, Sep. 2018.
[20] A. Tehreem, S. G. Khawaja, A. M. Khan, M. U. Akram, and S. A. Khan, ``Multiprocessor architecture for real-time applications using mean shift clustering," *J. Real-Time Image Process.*, pp. 1_14, Nov. 2017.