A Triangle Multi-Level Item-Based Collaborative Filtering Method That Improves Recommendations

Gharbi Alshammari[1], Stelios Kapetanakis[1,3], Nikolaos Polatidis[1], and Miltos Petridis[2]

[1] School of Computing, Engineering and Mathematics
University of Brighton, Moulsecoomb
Campus, Lewes Road, Brighton BN2 4GJ, UK
(g.alshammari, s.kapetanakis, n.Polatidis)@brighton.ac.uk
[2] Department of Computer Science
Middlesex University London,
The Burroughs, London NW4 4BT, UK
M.Petridis@mdx.ac.uk
[3] Gluru Research
71-91 Aldwych, London WC2B 4HN
stelios@gluru.co

Abstract. One of the most successful approaches that can provide a relevant recommendation in various domains is collaborative filtering. Although this approach has been widely applied, there are still limitation to be overcome in this research area. Accuracy is still one of the areas that needs to be improved. In addition, the rapid growth of information available online presents recommender systems with several challenges. More specifically, data sparsity and coverage affect the quality of the recommendations that can be provided. In this paper, we propose an item-based collaborative filtering (IBCF) approach with triangle similarity measures that take into account the length and angle of rating vectors between users and allow a positive and negative adjustments using a multi-level recommendation approach. We have improved the predictive accuracy and effectiveness of the proposed method, which outperforms all the compared methods in terms of the mean absolute error (MAE) and the root mean squared error(RMSE). We aimed to evaluate the proposed method by comparing our results with those of some popular similarity mesures using k-nearest neighbour (kNN) algorithms. We ran our experiment using three real dataset: MovieLens 100K, MovieLens 1 M and Yahoo! Movies.

Keywords: Collaborative filtering, recommender systems, triangle, multi-level, item-based

1 Introduction

Recommender systems help users and relevant items that meet their intersts to address the problem of information overload. Moreover, users have trouble

handling large volumes of information, and problems with cognitive and data sparsity when attempting to find appropriate information at the right time[4]. These systems play an important role in the growth of online information by filtering and recommending relevant items. The knowledge discovery approach can help with making a personalised recommendation by collecting a user's interests. Collaborative filtering (CF) is one of the most successful techniques for recommender systems [17].

Many collaborative filtering techniques have been proposed in different domains, such as e-commerce applications. Typically, elaborate approaches outperform the commonly-used k-nearest neighbour (kNN) baseline method in terms of accuracy, particularly for sparse datasets or in terms of scalability as they rely on offline pre-processing or model-building phases [6].

Most CF approaches analyse user ratings to determine the similarity between users and items. The similarity measure is important for finding accurate results in recommender system. However, it is challenging to determine distance measures in these systems in order to find similarities between users. Collaborative filtering is the most common applied algorithm through the kNN approach [11]. The key issue in this technique is how to calculate the similarity between users or items by finding similar shared interest. It is significantly rely on the rating aspect, which allow users to assign a high or low rating to a certain item based on their preference or dislike for it [13]. Many similarity measures have been adopted in recommender systems such as Pearson's Correlation Coefficient (PCC) [17] and Cosine [20] to provide recommendation based on the absolute ratings between users. Hence, modified similarity measures are one of the most important challenges to improve the prediction accuracy in recommender systems.

In this paper, we propose a new methods that utilises triangle similarity measures with a multi-level algorithm. We consider both the length and angle of the rating vectors between users, as well as the constraints that modify users similarity assigning these with different levels. The main contributions of this paper are as follows:

1. We proposed a new recommendation method that combines triangle similarity measures with a multi-level recommendation technique.
2. We ran extensive experiments to show its effectiveness based on three real datasets, conducting a comparison with a baseline and a state-of-the-art alternative.

2 Related Work

CF is the most popular technique for recommender systems. It has been widely implemented in different domains such as movies [15] and music [24] to generate recommendations. It is a method of information filtering that seeks to predict the rating that a user will give to a particular item based on a similarity matrix. CF provided a foundation for the first recommenders systems, which were used to help people make choices based on the opinions of other people [7]. The task is to make an automatic prediction by considering other similar users ratings

for an item. Therefore, the basic idea of CF is to find a user whose past rating behaviour is similar to that of the user the algorithm is currently trying to predict. This approach uses a kNN algorithm to calculate recommendations, and the main required data are the rating matrix and a function that computes similarity between users. An Item-based collaborative filtering (IBCF) technique looks into a set of items that the target users have rated and computes how similar they are to the target item; it then select the k most similar items. At the same time, the corresponding similarity are also computed [18]. Once the most similar items are found, the prediction is computed using a weighted average of the target users' ratings. Hence, there are two main aspects to be considered: similarity computation and prediction generation [2]. Basically, to compute the similarity between items, the first step is to determine users who have rated both items and who have the most similar items with similar ratings. Many different measures can be used to compute the similarity between items, such as Pearson's correlation, cosine, Jaccard and Triangle similarity. Of these methods, Pearson's has had the most successful applications, which is defined in Equation 1. Where $Sim_{a,b}$ is the similarity of users a and b, $r_{a,p}$ is the rating of user a for product p, $r_{b,p}$ is the rating of user b for product p and $\overline{r}a$, $\overline{r}b$ represent user's average ratings. P is the set of all products. However, In [21] the authors have proven that the triangle similarity outperforms Pearson correlation and shows improved results in IBCF.

$$\mathrm{Sim}_{a,b}^{PCC} = \frac{\sum_{P\in P}(r_{a,p}-\overline{r}a)(r_{b,p}-\overline{r}b)}{\sqrt{\sum_{P\in P}(r_{a,p}-\overline{r}a)^2}\sqrt{\sum_{P\in P}(r_{b,p}-\overline{r}b)^2}} \tag{1}$$

More recently, a combination of one or more methods called a hybrid recommender system has been applied to overcome the limitations of using one approach and obtain better results[5]. For instance, in [3], a hybrid case based reasoning approach was proposed to solve a long tail problem, which is items that have a few ratings by switching between collaborative filtering and content-based filtering. In addition, the authors in [12] implemented a hybrid recommender system that applied clustering technique and an artificial algae algorithm with a multi-level CF approach. However, co-rated items have been used for a problem solving in recommender systems to improve their predictive accuracy. Authors in [23] also introduced a hybrid approach for solving the problem of finding the rating of unrated items in a user-item matrix through a weighted combination of user-based and item-based collaborative filtering. These methods addressed the two major challenges of recommender systems, the accuracy of recommendations and sparsity of data, by simultaneously incorporating the correlation of users and items. In [22] the authors address a cold-start problem in user-based CF by considering both the distance between users and the co-rating of items using Jaccard factors. In [21], the authors proposed a new measure that integrates the triangle similarity approach with Jaccard similarity, which consider non co-rating users. The authors in [16] propose a multi-level constraint that improves the quality of a recommendation using PCC. Equation 2 considers the

similarity between users relying on PCC and co-rated items in different levels.

$$Sim_{a,b}^{PCC} = \left\{ \begin{array}{l} \mathrm{Sim}_{a,b}^{PCC} + x_1, if \frac{|I_a \cap I_b|}{T} \geq t1 \; and \; Sim_{i_j,i_q}^{PCC} \geq y \\ \mathrm{Sim}_{a,b}^{PCC} + x_2, \; if \frac{|I_a \cap I_b|}{T} < t1 \; and \frac{|I_a \cap I_b|}{T}| \geq t2 \; and \; Sim_{a,b}^{PCC} \geq y \\ \mathrm{Sim}_{a,b}^{PCC} + x_3, \; if \frac{|I_a \cap I_b|}{T} < t2 \; and \frac{|I_a \cap I_b|}{T}| \geq t3 \; and \; Sim_{a,b}^{PCC} \geq y \\ \mathrm{Sim}_{a,b}^{PCC} + x_4, \; if \frac{|I_a \cap I_b|}{T} < t3 \; and \frac{|I_a \cap I_b|}{T}| \geq t4 \; and \; Sim_{a,b}^{PCC} \geq y \\ 0, \quad otherwise \end{array} \right\}$$

(2)

3 The Proposed Method

The number of co-rated items reflects the degree of connection between users. For instance, a high number of co-rated items might indicate a high level of similarity. Traditional similarity metrics do not consider the number of co-rated items[19]. To solve this problem, a triangle similarity has been proposed by [21], which results in a significance improvement in accuracy when it is combined with co-rating. The triangle similarity is integrated with some constraints that apply a number of co-rated items.
In our approach, we apply a hybrid method that also adopts a multi-level CF approach, which enhances the similarity value of users that belong to certain categories and ignores the rest [16]. It enhances the process of kNN by finding a large margin within an application. The triangle similarity measure is defined as follows:

$$\mathrm{Sim}_{a,b}^{Tri} = 1 - \frac{\sqrt{\sum_{u \in c_{a,b}} (r_a - r_b)^2}}{\sqrt{\sum_{u \in c_{a,b}} r_a^2} + \sqrt{\sum_{u \in c_{a,b}} r_b^2}} \tag{3}$$

The value range is [0,1], where closer a value is to 1, the more similar they are. The triangle approach considers both the length of the vectors and the angle between them, so it is more reasonable than the angle-based cosine similarity.

For example if the two vectors $A = 5,5,5$ and $B = 1,1,1$ are given, then cosine similarity is 1. By contrast, the triangle similarity between them is 0.33.

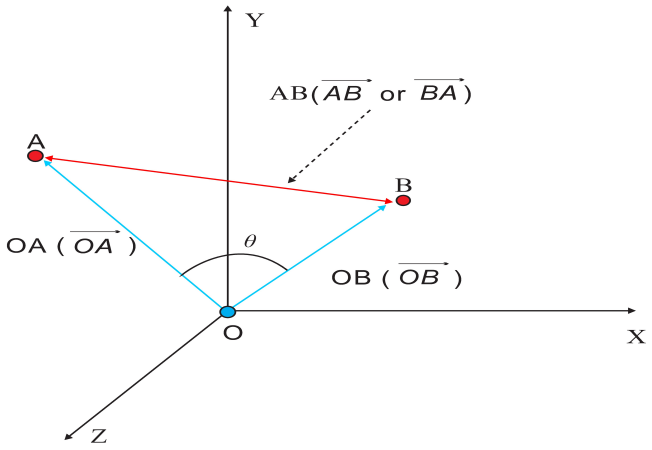

Fig. 1. A triangle in three-dimensional (3D) space [21].

$$Sim_{a,b}^{Proposed} = \begin{cases} \mathrm{Sim}_{a,b}^{Tri} + x_1, if \frac{|I_a \cap I_b|}{T} \geq t1 \; and \; Sim_{i_{j,iq}}^{Tri} \geq y \\ \mathrm{Sim}_{a,b}^{Tri} + x_2, \; if \frac{|I_a \cap I_b|}{T} < t1 \; and \frac{|I_a \cap I_b|}{T}| \geq t2 \; and \; Sim_{a,b}^{Tri} \geq y \\ \mathrm{Sim}_{a,b}^{Tri} + x_3, \; if \frac{|I_a \cap I_b|}{T} < t2 \; and \frac{|I_a \cap I_b|}{T}| \geq t3 \; and \; Sim_{a,b}^{Tri} \geq y \\ \mathrm{Sim}_{a,b}^{Tri} + x_4, \; if \frac{|I_a \cap I_b|}{T} < t3 \; and \frac{|I_a \cap I_b|}{T}| \geq t4 \; and \; Sim_{a,b}^{Tri} \geq y \\ 0, \quad otherwise \end{cases}$$

(4)

In the above equation, $sim_{a,b}$ denotes the similarity between user a and user b. T stands for the total number of co-rated items. $t1$, $t2$, $t3$ and $t4$ are the predefined threshold of co-rated items for user similarity $Sim_{a,b}^{Tri}$. We consider that $t1 = 50$, $t2 = 20$, $t3 = 10$ and $t4 = 5$. We took $x1 = 0.5$, $x2 = 0.375$, $x3 = 0.25$, $x4 = 0.125$ and $y = 0.33$.

Compare the similarity given by $Sim_{a,b}^{Tri}$ using y and the number of co-rated items. If it is less than a specified level t, then go to the next level and continue go to the next level until the right level is found. If all four levels are not found, then the similarity is equal to 0.

4 Experimental Evaluation

Given below are the results for the three datasets with different parameters. All algorithms were implemented in the Java programming language. In this experiment, k represents the number of nearest neighbours.

4.1 Real Dataset

We have run our experiment with three real datasets in order to compare the results with different parameters, such as the number of items and users. All datasets have been evaluated using cross-validation with 5 folds and K is the number of neighbours, specified to be equal to 3, 10, 30, 50 and 100.

MovieLens 100K: This is a real dataset that is publicly available. It uses a web-based research recommender system that was conducted from September 1996 to April 1998. It contains 943 users and 1,682 movies. Each user has rated at least 20 movies. It contains 100,000 ratings, all of which are in a range between 1 and 5. The three main features are [UserID], [MovieID] and [Rating][8].

MovieLens 1M: This dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6040 users. The University of Minnesota created an online movie recommendation system, and its items are rated by users who joined MovieLens in 2000. All ratings are in a scale between 1 and 5. This dataset is also publicly available for running offline experiments and is widely used for collaborative filtering recommender systems [8].

Yahoo! Movies: This is a dataset obtained from Yahoo Labs under license. It contains 7,642 users, 11,915 movies and 211,111 ratings. The rating scale is between 1 and 5[1].

4.2 Comparison

We ran the following IBCF algorithms to make a comparison between the following methods. All the methods used are detailed in the following sections:

PCC: In this method, the statistical correlation between the similar ratings of two users is calculated to find users that are the closest to a particular user. The output will be a value between -1 and 1; 1 is a totally positive correlation, 0 indicates that there is no correlation and -1 is a totally negative correlation.

Multi-level CF: This is a method that calculates the statistical correlation between the similar ratings of two users to find users that are the closest to a particular user. The output will be a value between -1 and 1; 1 is a totally positive correlation, 0 indicates that there is no correlation and -1 is a totally negative correlation.
The above two methods are compared in our proposed method. Our method considers both the length and angle of the rating vectors between users. Multi-level approach also considers the right level for each user after calculating the triangle similarity and compare it with a specified threshold. In addition, in each level, a certain constraint is conducted to modify the similarity between certain users who share similar items.

4.3 Evaluation Metrics

Recommender system researchers have applied different measures to evaluate the quality of proposed recommendation algorithms [10]. Since 1994 [17], most of the empirical studies examining recommender systems have focused on appraising the accuracy of these systems using different methods [9]. Appraisals of accuracy are useful for evaluating the quality of a system and its ability to forecast the rating for a particular item. Predictive accuracy measurement metrics are widely used by the research community in CF, which measures the similarity between true user ratings and recommender system predicted ratings. Hence, we apply both the mean absolute error (MAE) and the root mean squared error ($RMSE$) to measure the performance of the proposed methods and evaluate their prediction accuracy compared with other recommendation techniques. The MAE is defined in Equation 5 and the RMSE is defined in Equation 6.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \tag{6}$$

In the above equations, p_i is the predicted rating, and r_i is the actual rating. It should be considered that lower values provide better result.

4.4 Experiment and Results

The MAE results in Fig 2 show that when the number of neighbours is small, the prediction is significantly improved. For example. when $k = 3$ the PCC is = 1.002 ; in multi-level CF, is 0.83 whereas in our proposed method, it is = 0.82. In addition, our method outperforms other methods in all cases, even when the number of neighbourhood is high. In Fig 3, we can see that the results of using the RMSE are significant in our method at $K = 30$ and 50.

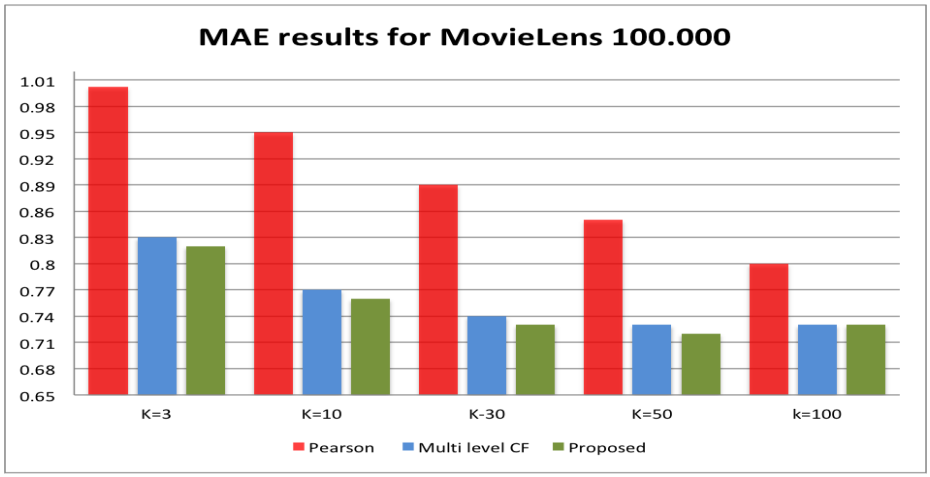

Fig. 2. MAE results for the MovieLens 100K dataset.

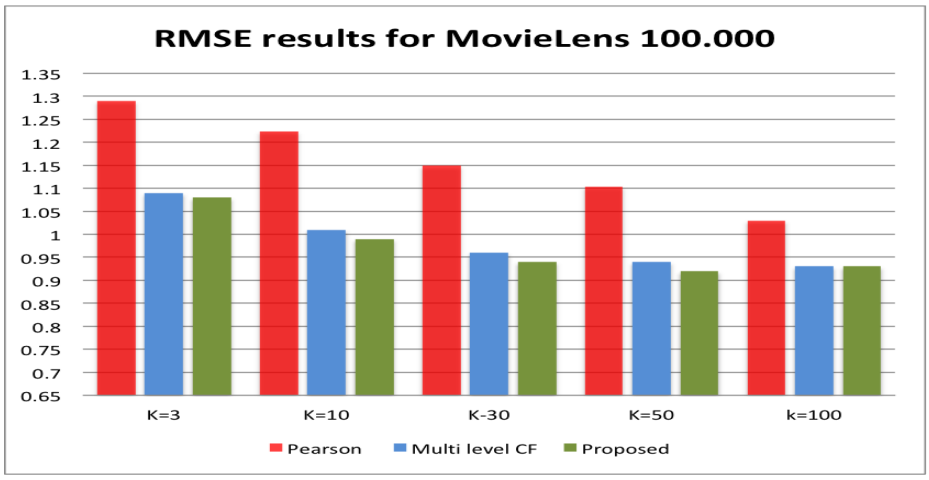

Fig. 3. RMSE results for the MovieLens 100K dataset.

Figs 4 and 5 show the MAE and RMSE results using MovieLens 1M ratings. In these figures, the improvement of our method compared with other methods is clear. However, when the number of neighbourhood is greater than 50, the multi-level CF value becomes close or equal to the result of our proposed method, such as $k = 100$. Overall, our proposed method achieves the minimal MAE in all cases.

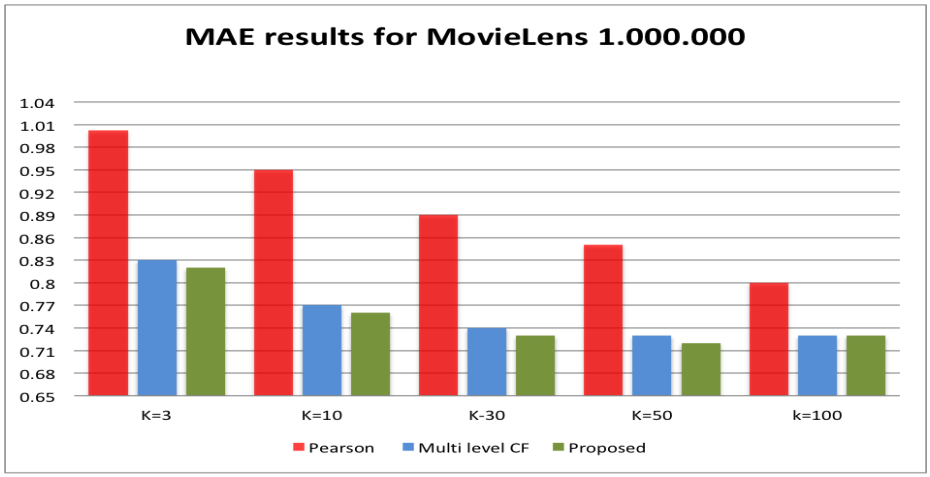

Fig. 4. MAE results for the MovieLens 1M dataset.

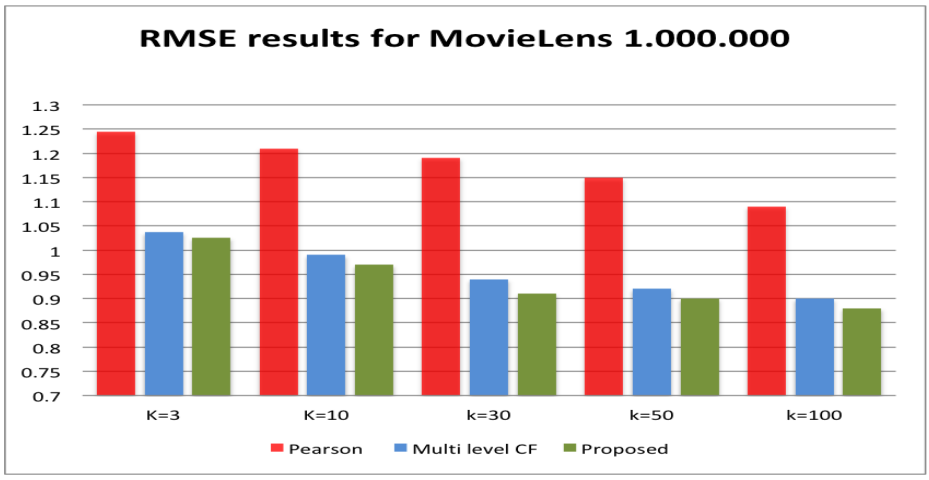

Fig. 5. RMSE results for the MovieLens 1M dataset.

For the Yahoo! Movies dataset shown in Figs 6 and 7, we can see a significant change in all examined k value. For example, when $k = 3$, the baseline value is 0.85, the multi-level is 0.79 and the value in our method is 0.76. When $k = 100$ it can be seen that the difference between the three methods is slightly smaller but the proposed method still has the best results.

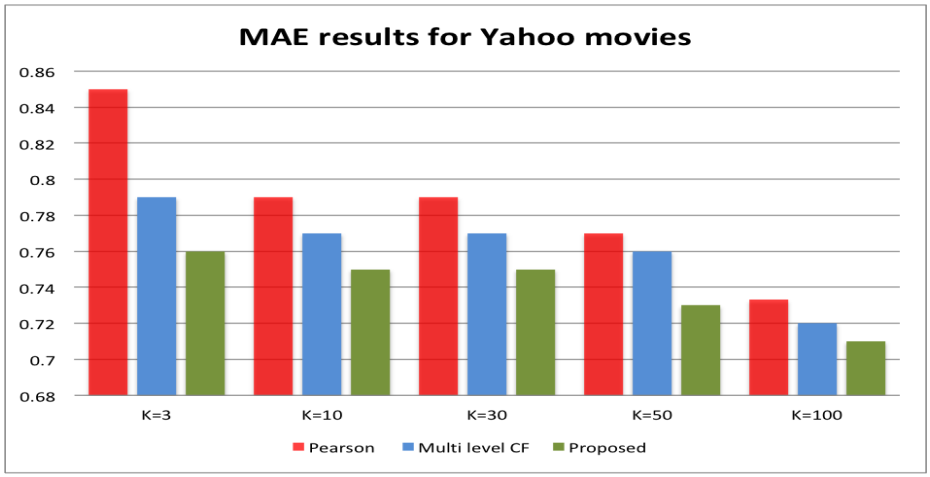

Fig. 6. MAE results for Yahoo! Movies dataset.

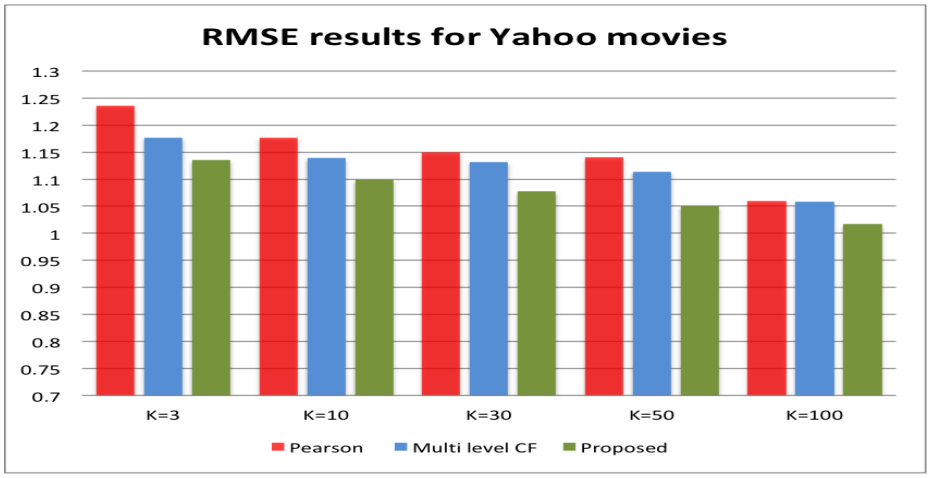

Fig. 7. RMSE results for Yahoo! Movies dataset

5 Discussion

In this paper we presented a new collaborative filtering method based on Triangle Similarity and multi-layer, multi-level similarity features. The proposed method has been experimentally evaluated using three real datasets and well known prediction accuracy metrics with the results being promising. In the MovieLens datasets it is shown that the proposed approach marginally outperforms the

alternative, while in the Yahoo movies dataset the difference is higher between our method and the alternatives. Furthermore, the prediction error becomes smaller for each of the methods and in every dataset as the neighborhood is growing with the difference between our method and the alternative being similar in all cases. The proposed method outperforms the Pearson baseline and the multi-level CF state-of-the-art approach in all three datasets using both MAE and RMSE.

Although, the results are promising, they follow an instance-base recommendation approach, something that is promising in the short-term. However, they can pose severe limitations to richer, context-oriented user journeys. To overcome this limitation and as part of our future work we aim to use a Recurrent Neural Network (RNN) approach for pattern matching in collaborative filtering. RNNs are strong candidates for sequence matching and identification due to their consecutive structure and parsing of prev-current-next sequence entries for a given domain. RNNs are able to predict the next item in given several traces of the previous ones [14]. We propose an RNN approach compared to a possible probabilistic alternative (e.g. Hidden Markov Chains(HMC)) since it turn to minimise the number of iterations (only two data scans compared to multiple ones in the case of a HMC).
A single RNN will predict its expected output: y based on its hidden internal state: h_0 .. h_{t-1} with an example can be seen in figure 8 below:

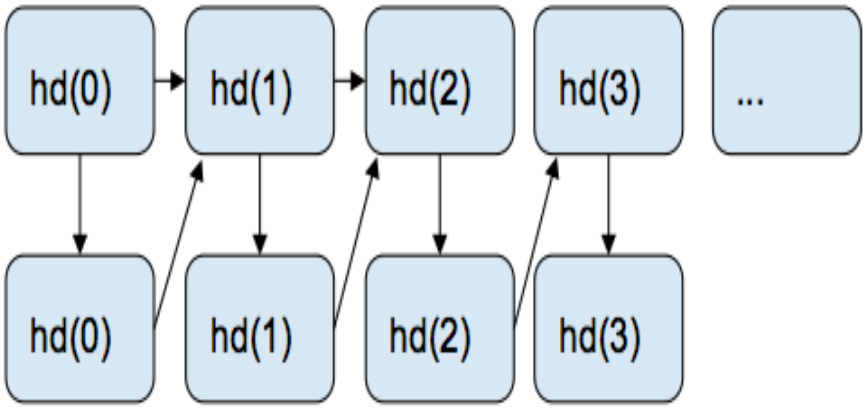

Fig. 8. RNN internal sequence prediction.

We assume sequences of (u_t, r_t) pairs where u_t is the user at time t and r_t is the desired recommendation. Such a pair is signal sequence and our dataset comprises several of them. For a recurrent network there is an additional input, a hidden state from the previous time step. We will represent it as h_{t-1} . A

predicted output can be represented as r'_t. We define weight matrices as W_{hu} for the input to hidden layer weights, W_{hh} for hidden to hidden layer weights and W_{rh} for the hidden to output weights. From the above definitions We define a recurrent network as shown in equations 7 and 8:

$$h_t = \sigma_h(W_{hh}h_t - 1 + W_{hu}u_t + b_h) \tag{7}$$

$$r'_t = \sigma_r(W_{rh}h_t + b_r) \tag{8}$$

Equations 7 and 8 can be applied to define a propagation function. To learn the weights we will generalise through time. Sequence prediction seems the first step towards a continuous time learning that should allow us to create the multiple levels dynamically, thus reducing the possibility of misbehavior.

6 Conclusions and Future Work

In this paper, we proposed a novel IBCF method based on triangle similarity and multiple similarity levels. The proposed method has been experimentally evaluated using three real datasets, along with a comparison to the traditional baseline PCC and to the state of the art multi-level CF. The results clearly show that the quality of the recommendations is improved in all cases and that our proposed method outperforms the alternatives.

Recommender systems have been widely used in different domains, including e-commerce and social media, so it is important to provide the best recommendations possible. Therefore, we have concentrated the best recommendations possible by reducing the prediction error, which results in high-quality recommendations. Furthermore, in the future, we aim to improve our method by making it dynamic and by performing a broader experimental evaluation.

References

1. Yahoo! research webscope movie data set. version1.0, http://research.yahoo.com/
2. Aggarwal, C.C., et al.: Recommender systems. Springer (2016)
3. Alshammari, G., Jorro-Aragoneses, J.L., Kapetanakis, S., Petridis, M., Recio-García, J.A., Díaz-Agudo, B.: A hybrid cbr approach for the long tail problem in recommender systems. In: International Conference on Case-Based Reasoning. pp. 35–45. Springer (2017)
4. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Knowledge-Based Systems Recommender systems survey 46, 109–132 (2013)
5. Burke, R.: Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction 12(4), 331–370 (2002)
6. Gedikli, F., Jannach, D.: Recommending based on rating frequencies: Accurate enough? In: Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at UMAP10 (ITWP10). pp. 65–70 (2010)
7. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)

8. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5(4), 19 (2016)
9. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 230–237. ACM (1999)
10. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
11. Jeong, B., Lee, J., Cho, H.: Improving memory-based collaborative filtering via similarity updating and prediction modulation. Information Sciences 180(5), 602–612 (2010)
12. Katarya, R., Verma, O.P.: Effectual recommendations using artificial algae algorithm and fuzzy c-mean. Swarm and Evolutionary Computation 36, 52–61 (2017)
13. Konstan, J.A., Riedl, J.: Recommender systems: from algorithms to user experience. User modeling and user-adapted interaction 22(1-2), 101–123 (2012)
14. Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
15. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: Movielens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th international conference on Intelligent user interfaces. pp. 263–266. ACM (2003)
16. Polatidis, N., Georgiadis, C.K.: A multi-level collaborative filtering method that improves recommendations. Expert Systems with Applications 48, 100–110 (2016)
17. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. pp. 175–186. ACM (1994)
18. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295. ACM (2001)
19. Shen, K., Liu, Y., Zhang, Z.: Modified similarity algorithm for collaborative filtering. In: International Conference on Knowledge Management in Organizations. pp. 378–385. Springer (2017)
20. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACM Computing Surveys (CSUR) 47(1), 3 (2014)
21. Sun, S.B., Zhang, Z.H., Dong, X.L., Zhang, H.R., Li, T.J., Zhang, L., Min, F.: Integrating triangle and jaccard similarities for recommendation. PloS one 12(8), e0183570 (2017)
22. Tan, Z., He, L.: An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. IEEE Access 5, 27211–27228 (2017)
23. Wei, S., Zheng, X., Chen, D., Chen, C.: Electronic Commerce Research and Applications A hybrid approach for movie recommendation via tags and ratings q 18, 83–94 (2016)
24. Yoshii, K., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE Transactions on Audio, Speech, and Language Processing 16(2), 435–447 (2008)