# A switching multi-level method for the long tail recommendation problem

Gharbi Alshammari [a], Jose L. Jorro-Aragoneses [b], Nikolaos Polatidis [a,*], Stelios Kapetanakis [a], Elias Pimenidis [c] and Miltos Petridis [d]

[a] *School of Computing, Engineering and Mathematics, University of Brighton, Lewes Road, BN2 4GJ, Brighton, United Kingdom*
*E-mail: G.Alshammari, N.Polatidis, S.Kapetanakis@Brighton.ac.uk*

[b] *Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Av. Séneca, 28040, Madrid, Spain*
*E-mail:jljorro@ucm.es*

[c] *Department of Computer Science and Creative Technologies, University of the West of England, Frenchay Campus, FS16 1QY, Bristol, United Kingdom*
*E-mail:Elias.Pimenidis@uwe.ac.uk*

[d] *Department of Computer Science, Middlesex University,The Burroughs, NW4 4BT, London, United Kingdom*
*E-mail:M.Petridis@mdx.ac.uk*

**Abstract.** Recommender systems are decision support systems that play an important part in generating a list of product or service recommendations for users based on the past experiences and interactions. The most popular recommendation method is Collaborative Filtering (CF) that is based on the users' rating history to generate the recommendation. Although, recommender systems have been applied successfully in different areas such as e-Commerce and Social Networks, the popularity bias is still one of the challenges that needs to be further researched. Therefore, we propose a multi-level method that is based on a switching approach which solves the long tail recommendation problem (LTRP) when CF fails to find the target case. We have evaluated our method using two public datasets and the results show that it outperforms a number of bases lines and state-of-the-art alternatives with a further reduce of the recommendation error rates for items found in the long tail.

Keywords: Recommender Systems, Collaborative Filtering, Switching, Multi-level, Long Tail Recommendations

## 1. Introduction

Recommender systems (RS) are decision support systems well known for their use in filtering and finding the relevant products on the web, thus solving the informtion overload problem. RS can make a huge impact on both sides: (1) increasing the sales of a business and (2) reduce the burden of users by finding and recommending interesting items. These recommendations rely on user interaction and behaviour tracking and analysis using artificial intelligence approaches [1].

Collaborative filtering (CF) is the most successful technique for recommender system. Given a set of users, items and ratings, CF will suggest items to a particular user based on common previous ratings with other users. The main task of CF is to predict the rating of a certain item that might meet the user interest based on common previous user ratings. The rating is the most important input in CF, which can be gathered explicitly or implicitly[2]. It works on the idea that recommending items based on the similarity between users and it was first proposed in the mid

1990 using the most common classification model: K-nearest neighbours (kNN). kNN main advantage of use is that it is simple to implement. Secondly, the efficiency point which needs no costly steps to train the model. Thus, it becomes popular among researchers [3]. However, CF suffers from the long tail problem, which affects the accuracy of the recommendations [4]. The key issue in this technique is how to calculate the similarity between users or items by finding similar shared interest. It relies on the ratings, which allow users to assign a high or low rating to a certain item based on their preference or dislike for it [5].

On the other hand, content based filtering (CBF) is another recommendation technique that considers the features of the items to find the similarity between them. For example, in user terms the user profile is representing the content of the items that have been liked/rated to reflect the user interests and preferences. Therefore, to make relevant recommendations that match against a user profile, a similarity measure is adopted that calculates a similarity value that is close to the user profile.

Many similarity measures have been adopted in recommender systems such as Pearson's Correlation Coefficient (PCC) [2] and Cosine [6] to provide recommendations based on the absolute values of the ratings between users. Thus, modified similarity measures considered an important research area with an aim to improve the prediction accuracy.

Regarding hybrid recommender systems, the author in [7] proposed a different way of using two or more techniques through seven hybridization methods including: weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level. The main goal to combine the aforementioned methods is to achieve higher quality of recommendations by providing more reliable and accurate results compared to when one method is used. The authors presented one category of the hybrid recommendation that called EntreeC. It is a restaurant recommender system that combines CBR and Collaborative filtering as a cascade method using the knowledge representation as a first step to rank the similar users based on their interest. Then, CF is employed among those users.

Many recommender systems algorithms have considered the popular items or items with the highest rating which are called popularity based recommender systems. For example, in news when you read a daily news website, it will recommend you the popular news based on the most popular news article according to reading frequency. However, the challenge comes when the items are new to the system or have not gained enough rating to become popular among others. This issue is really essential to consider less known items more than the popular one since it can add the serendipity to the users. These items are belonging to the problem of the long tail as it is introduced in [8]. Hence, those items should be considered and the method is able to suggest the relevant one in the tail. For example, the authors in [9] presented an item weighting approach that filters the items in the long tail and recommend them within the top ranked items. Considering the importance of the long tail recommendation problem, in this paper, we propose a novel method that integrates the multi-level method with the switching hybrid system. The main contributions of our method are as follow:

- We proposed a novel recommendation method that applies a switching approach between CF and CBF using a multiple-level method that improves the prediction accuracy when recommending the items in the long tail.
- We examine the proposed method through a comprehensive experiment on two public datasets using two different evaluation approaches to show the quality of the proposed method, conducting a comparison with the baseline methods and a state-of-the-art alternative.

The rest of the paper is organized as follows: Section 2 contains the related work, section 3 presents the proposed method, section 4 delivers the experimental evaluation, section 5 contains the discussion and section 6 describes the conclusions and future work parts.

## 2. Related Work

A major challenge in recommender systems is to provide a list with high quality recommendations to the users. This challenge is mostly managed by first finding a probability of the user to what to watch or purchase through rating prediction, then at the second step a ranking of the items that have a high impact follows. In the literature many works in this area focus on the two most applied methods: CF and CBF. CF relies on the rating similarity that is based on the assumptions the similar users rate the similar items, which can help predict unseen items [10]. On the other hand, CBF is based on the similarity of items features for example: genres or some text which represent the item using information retrieval and filtering techniques, e.g. the term Frequency-Inverse Item Frequency (TF-IDF). However, the effectiveness of both methods is limited when presented individually. Thus, hybrid recommender systema was proposed as a term in 2002 by [7] to solve the limitation of each method by using two or more methods. The recommendations are generated to a specific user through a prediction using a similarity function that calculates how two users are similar. Then, the classification model estimates and identifies who is the closest one that can help calculating the predicted value. One of the most widely used classifier is the kNN which presents the most similar user utilizing a pre-defined number of user with similar ratings which are usually refered as nearest neighborous and defined as k. In addition, the most popular measures that have been utilized in the literature are Pearson correlation coefficient (PCC), Euclidean distance and Cosine similarity [11]. Pearson similarity is the most known and used and 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.

$$Sim_{a,b}^{\mathrm{PCC}} = \frac{\sum\limits_{p\in P}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum\limits_{p\in P}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum\limits_{p\in P}(r_{b,p}-\bar{r}_b)^2}} \quad (1)$$

Recently, many authors presented a modification to the similarity function to improve the CF recommendation [12,13,14,15]. For example, a multi-level method was proposed by the authors in [12] that utilize a nbumber of constraints that enhance the Pearson correlation coefficient (PCC) similarity value of users who belonged to specified categories based on the number of co-rated items and the minimun PCC value between them. This methdod is explained in more detail towards the end of this section. In addition, in [15], another multi-level approach was presented which is similar to the aforementioned but the triangle similarity was used, which improves the prediction accuracy even more. In [14], the cosine similarity was modified using co-rated items as an adjusted factor to improve the similarity. 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[7]. For instance, in [16], a hybrid case based reasoning approach was proposed to solve the long tail problem, which basically refers to items that have few ratings, by switching between collaborative filtering and content-based filtering. In addition, the authors in [17] 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 [18] 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 the recommendations and the sparsity of data, by simultaneously incorporating the correlation of users and items. In [19] 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 similarity. In [20], the authors proposed a new measure that integrates the triangle similarity approach with Jaccard similarity. The authors in [12] proposed 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 Where, $sim_{a,b}$ 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.

$$\mathrm{Sim}_{a,b}^{\mathrm{PCC}} = \begin{cases} \mathrm{Sim}_{a,b}^{\mathrm{PCC}} + x_1 & \text{if} \quad \frac{|I_a \cap I_b|}{T} \geq t_1 \text{and} \mathrm{Sim}_{ij,iq}^{\mathrm{PCC}} \geq y \\ \mathrm{Sim}_{a,b}^{\mathrm{PCC}} + x_2 & \text{if} t_2 \leq \frac{|I_a \cap I_b|}{T} < t_1 \text{and} \mathrm{Sim}_{a,b}^{\mathrm{PCC}} \geq y \\ \mathrm{Sim}_{a,b}^{\mathrm{PCC}} + x_3 & \text{if} t_3 \leq \frac{|I_a \cap I_b|}{T} < t_2 \text{and} \mathrm{Sim}_{a,b}^{\mathrm{PCC}} \geq y \\ \mathrm{Sim}_{a,b}^{\mathrm{PCC}} + x_4 & \text{if} t_4 \leq \frac{|I_a \cap I_b|}{T} < t_3 \text{and} \mathrm{Sim}_{a,b}^{\mathrm{PCC}} \geq y \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The long tail recommendation problem (LTRP) is a major challenge in recommender system that refers to less popular items[8]. In the literature, a number of ways has been presented to solve this problem with the majority based on a pre-processing technique such as clustering or by dividing the data into groups (head and tail)[21,22,23]. For example, the authors in [24] describe a clustering technique that boosts items belonging to the long tail. In [21], an item clustering approach was proposed based on association rule mining. In addition, matrix factorization was proposed in [25] to evaluate the performance of the recommendation of items in the long tail. Graph-based algorithms have also been proposed in [22] for the long tail recommendation by using user-item information along with undirected edge weighted graphs for long tail item recommendation. In [26] a case base reasoning method presented and showed that the recommendations were based on unknown artists and tracks. The proposed system in that study could identify whether an item resided in the long tail and if it were attempting to improve its provided meta-data through the addition of tag knowledge information. However, most of the existing algorithms require additional processing to solve the long tail problem and some algorithms decrease the accuracy when recommending items in the long tail [22] whereas in our method the accuracy is increased and no additional processing or information is required.

## 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 indicates a higher level of similarity. Traditional similarity metrics do not consider the number of co-rated items[14]. To solve the LTRP, a switching hybrid system is proposed that utilize the multi-level similarity method [12].

In Figure 1, it is shown that our proposed architecture switches between two different techniques: a collaborative filtering component that calculates the predicted rating based on other similar users and content-based filtering. The content-based component calculates the predicted rating using other similar items that the user has rated in the past. In our method, we apply a hybrid approach that adopts the multi-level CF approach, which enhances the similarity value of users that belong to certain categories and ignores the rest [12] as shown in equation 2. The system receives a query $(Q)$ that identifies the target user $(u)$ and item $(m)$. Ratings are in a scale from 1 to 5 and the goal is to compute the estimated rating for the item $r(m,u)'$

$$Q = < u, m > \tag{3}$$

The first step in the system is to decide which method is more effective in correctly calculating the rating prediction. This decision is based on the number of ratings received by the target item. In order to make

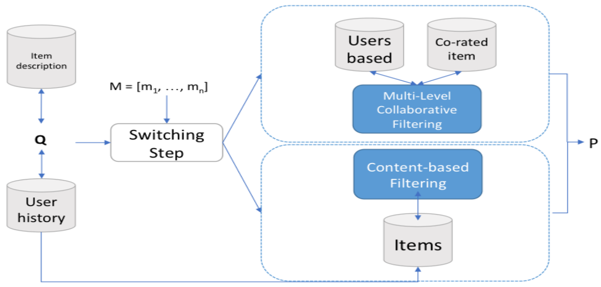


Fig. 1. A switching multi-level recommender system architecture.

this decision, the system computes a vector $(R_m)$ that represents the ratings of a concrete item $m$ and user $j$.

$$R_m = \langle m, r \rangle_m = (\langle m, r(m, u_1) \rangle, \ldots, \langle m, r(m, u_j) \rangle) \tag{4}$$

In this first step, the system obtains the number $|R_m|$ of ratings that the query item $(m)$ has. Then, it compares this value with a threshold constant $(\delta)$. If the number of ratings of $m$ is higher than $\delta$, then this item is not in the long tail and the multi-level collaborative filtering method can be used. On the other hand, if the number of ratings is lower than $\delta$, the system can not find similar users that rated this movie, due to the fact that the system does not have a sufficient number of ratings. Therefore, in this case the system switches to the content based method. To summarise, the main steps of our proposed switching method are:

1: Obtain the number of ratings that a target item has received.

2: If it is larger than the $\delta$ then CF is used otherwise CBF is utilised.

## 4. Experimental Evaluation

### 4.1. Comparison

We ran the proposed switching hybrid method and compared with the baseline CF using Euclidean and Pearson similarity, with the CBF and with the method in [16].

#### 4.1.1. User-based CF

This model is used to calculate the rating prediction based on the ratings of similar users $u'$. The model computes a list with all items rated by the user $(R_u)$

and compares the list in order to obtain the user similarity. In this model, the user obtained by the kNN algorithm must have rated the target item $m$. Then, the similarity configured using two most popular similarity function. First, the Euclidean distance:

$$Sim_{Euc}(R_u, R_{u'}) = 1 - \sqrt{\sum_{m=0}^{|M|} (r(m,u) - r(m,u'))^2} \tag{5}$$

where

$$M = R_u \cap R_{u'} \tag{6}$$

The other similarity function explored in the experimental evaluation is the Pearson correlation that is defined in equation 1:

When the system has retrieved the $k$ most similar users that have rated the target item $m$, it calculates the rating prediction using the other ratings derived from these users. This prediction is calculated with the weighted average of the rating and the similarity measure.

$$r(m,u)' = \frac{\sum_{i=0}^{k} r_i(m,u') * sim_i(R_u, R_{u'})}{\sum_{i=0}^{k} sim_i(R_u, R_{u'})} \tag{7}$$

Finally, $r(m,u)'$ is the result returned by this module as the rating prediction. We will now explain the second step.

#### *4.1.2. Content based filtering based on user history*

This model is based on the statistical average of ratings per genre defined in each user profile. It calculates the predicted rating using other similar movies that the user has rated in the past. This system creates a personal case base for the target user. Each case $(C_{CB})$ contains a list of genres that describes the movie

$$C_{CB} = < u, m, G_m > \tag{8}$$

$$G_m = \{g_1, \ldots, g_i, \ldots, g_n\} \tag{9}$$

Now, given a query, movies are compared according to the number of common genres using Jaccard similarity defined in 10.

$$sim(m,m') = \frac{G_m \cap G'_m}{G_m \cup G'_m} \tag{10}$$

The CBF is calculated using the $k$ most similar movies, the rating prediction is calculated using Equation 7.

#### *4.1.3. Multi-Level*

It is based on multiple levels, from top to bottom, with each of these levels having a number of constraints that is defined in equation 2. Where, $sim_{a,b}$ 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. 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.

#### *4.1.4. ICCBR17*

This model employs a switching method between the CF and CBF using a constraint that specifies whether to apply a CF or CBF based on the number of rating received with regards to the target item.

#### *4.1.5. Details of the experiments*

Our proposed method is based on a switching method that uses the multi-level algorithm with a number of constraints. The constraints assist the method to provide recommendations with lower prediction errors between users that have more common items and a PCC similarity value above a certain threshold, which is something that is not available in the other methods. However, it should be noted that if any of the constraints of the level is not satisfied then the similarity value between the users will be set to zero $Sim_{a,b}^{PCC}$, which is different from equation 2. The Experiments were conducted with an aim of comparison of the proposed method results against the baseline CF, CBF algorithms and the ICCBR17 method. The experiments are based on two publicly available MovieLens datasets and the most known accuracy measures for the error predictive accuracy in recommender systems: the mean absolute error (MAE) and the root mean squared error (RMSE). In the sections below the results are presented for two real datasets based on two different training and testing percentage to evaluate the results and compare it with the other methods. In this experiment, $k$ represents the number of neighbours, specified to be equal to 3, 5, 10, 20 and 30.

### 4.2. MovieLens 100K

This is a real dataset that is publicly available. It uses a web-based research recommender system. 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]. The version used for this dataset is the newest available version.

### 4.3. MovieLens 1m

This is a real dataset that is publicly available. It uses a web-based research recommender system. It contains 6000 users and 4000 movies. Each user has rated at least 20 movies. It contains 1 million ratings, all of which are in a range between 1 and 5. The three main features are [UserID], [MovieID] and [Rating]. From this dataset we have the first 500 users as found on its database.

### 4.4. Evaluation Metrics

Recommender system researchers have applied different measures to evaluate the quality of proposed recommendation algorithms [27]. Since 1994 [2], most of the empirical studies examining recommender systems have focused on appraising the accuracy of these systems using different methods [28]. 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. Therefore, we employed both $(MAE)$ and $(RMSE)$ to evaluated the performance of the proposed method and validate their prediction accuracy compared with other recommendation techniques. MAE is defined in Equation 11 and RMSE is defined in Equation 12.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \tag{11}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \tag{12}$$

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 results.

Additionally we have used the improvement rate based on MAE and RMSE respectively. Equation 13 defines the improvement rate for MAE and equation 14 defines the improvement rate for RMSE. $MAE_{base}$ represents the error rate for the compared algorithms that is conducted as a baseline. On the other hand, $MAE_{Switching}$ represents the error rate value for our proposed algorithm. $RMSE_{base}$ represents the error rate for the compared algorithms that is conducted as a baseline. On the other hand, $RMSE_{Switching}$ represents the error rate value for our proposed algorithm.

$$IR = \frac{MAEbase - MAE_{Switching}}{MAEbase} \tag{13}$$

$$IR = \frac{RMSE_{base} - RMSE_{Switching}}{RMSE_{base}} \tag{14}$$

### 4.5. Results

#### 4.5.1. Movielens 100K dataset

Figures 2, 3, 4 and 5 show the MAE and the RMSE error rates respectively across MovieLens 100K dataset using the aforementioned predictive methods utilizing 70 % for training and 30 % testing in figures 2 and 3. Additionally, 60% for training and 40% for testing in figures 4 and 5. We have adopted an approach where we have evaluated using 70:30 and 60:40 for 3 times and the values represent the averages. It is shown that our proposed method outperforms all the other compared recommendation methods. It can be seen clearly that when the number of neighbours is smaller, for example, when $k$ = 3, 5 and 10, the improvement is very significant. On the other hand, when the $k$ is getting higher we still have an improvement in our method but it is less effective. Additionally to that, figures 6 and 7 present the improvement rates for MAE and RMSE respectively for the MovieLens 100.000 dataset. It is shown in the comparison that the prediction accuracy is improved when the proposed method is being used both against the baselines and against the ICCBR17.

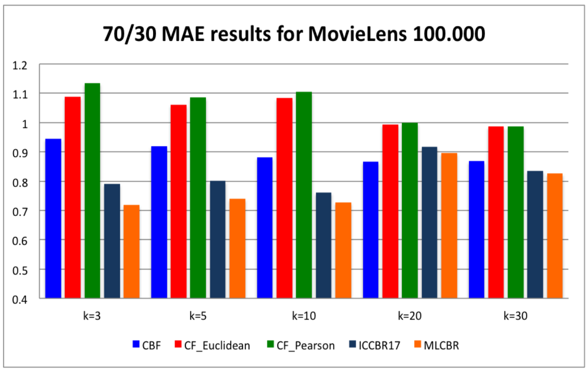


Fig. 2. MAE result for MovieLens 100.000 dataset using 70/30 test

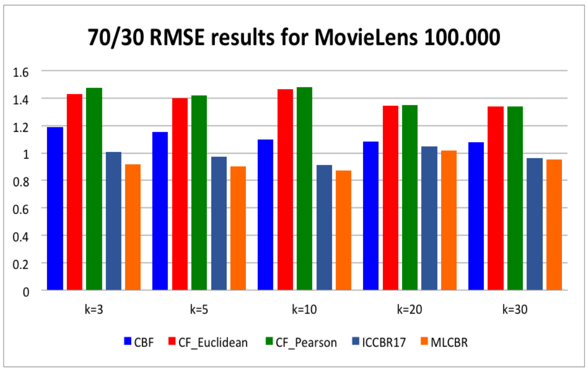


Fig. 3. RMSE result for MovieLens 100.000 dataset using 70/30 test

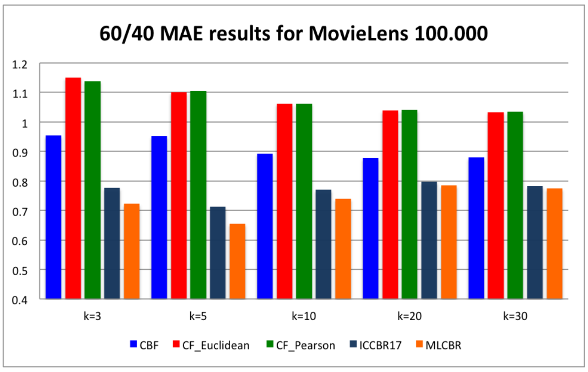


Fig. 4. RMSE result for MovieLens 100.000 dataset using 60/40 test

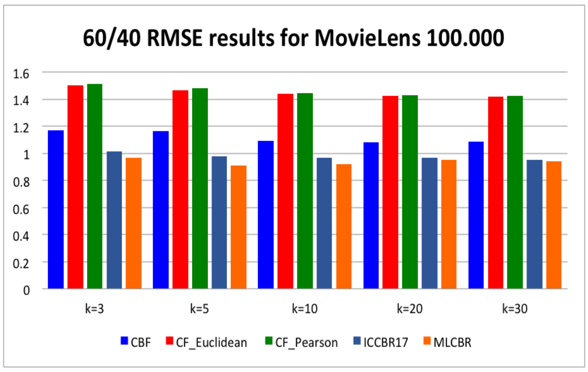


Fig. 5. RMSE result for MovieLens 100.000 dataset using 60/40 test

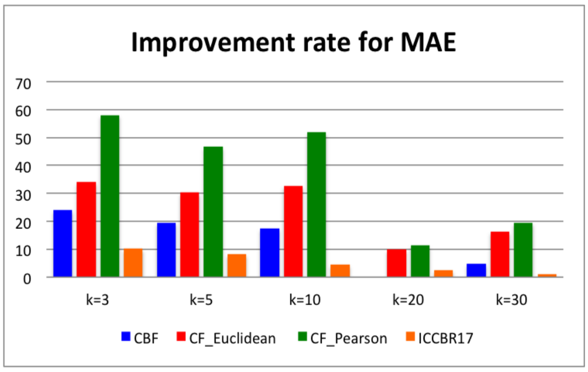


Fig. 6. Improvement rate for MAE based on MovieLens 100.000.

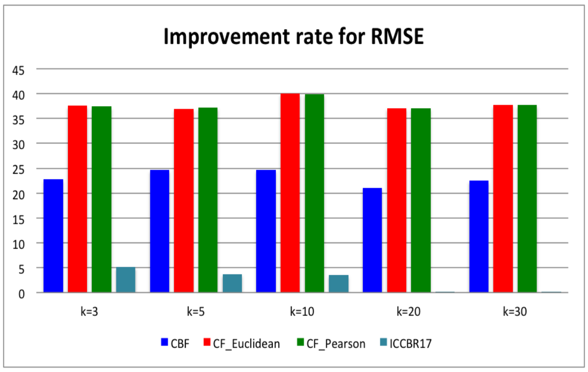


Fig. 7. Improvement rate for RMSE based on MovieLens 100.000.

*4.5.2. Movielens 1 million dataset*

Figures 8, 9, 10 and 11 show the MAE and the RMSE error rates respectively across MovieLens 1m dataset using the aforementioned predictive methods utilizing 70 % for training and 30 % testing in figures 8 and 9. In addition, figures 10 and 11 use 60% for training and 40% for testing. It is shown that our proposed method outperforms all the other compared recommendation methods. It can be seen clearly that when the number of neighbours is smaller, for example, when $k$ = 3, 5 and 10, the improvement is very significant. On the other hand, when the $k$ is getting higher we still have an improvement in our method but it is less effective. Additionally to that, figures 12 and 13 present the improvement rates for MAE and RMSE respectively for the MovieLens 1m dataset. It is shown in the comparison that the prediction accuracy is improved when the proposed method is being used both against the baselines and against the ICCBR17.

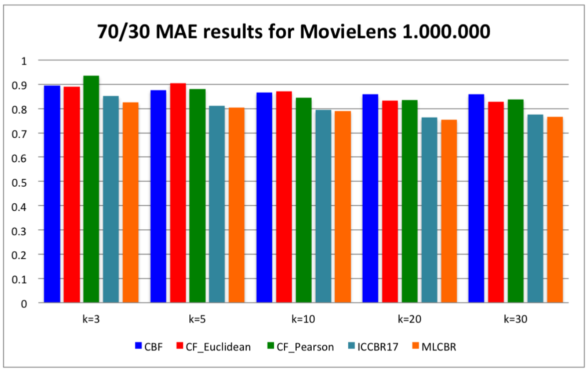


Fig. 8. MAE result for MovieLens 1m dataset using 70/30 test

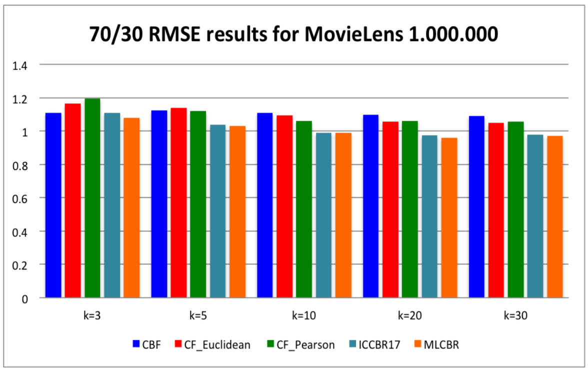


Fig. 9. RMSE result for MovieLens 1m dataset using 70/30 test

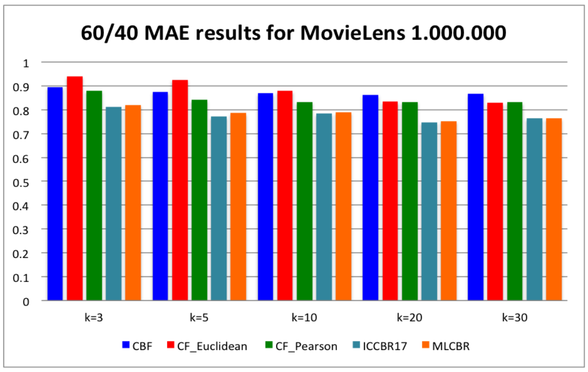


Fig. 10. MAE result for MovieLens 1n dataset using 60/40 test

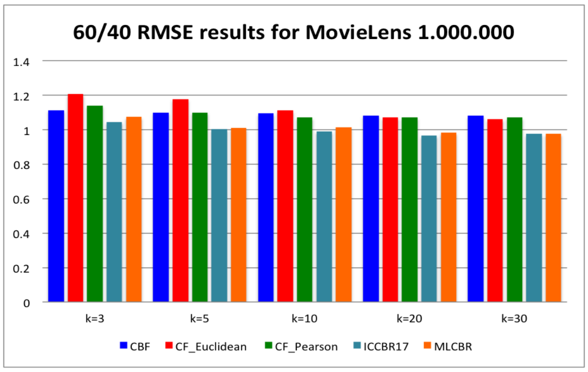


Fig. 11. RMSE result for MovieLens 1m dataset using 60/40 test

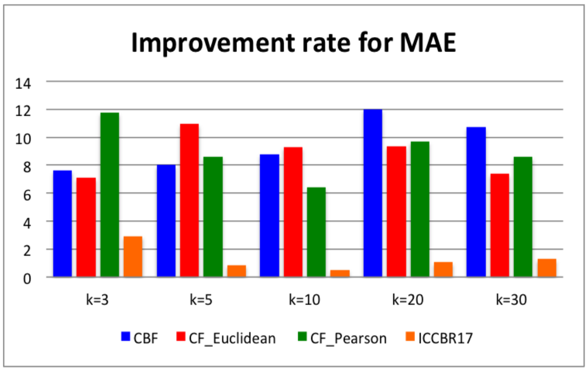


Fig. 12. Improvement rate for MAE based on MovieLens 1m

## 5. Discussion

The long tail recommendation problem becomes a problem and opportunity at the same time when a point has been reached within a business where offering the same popular products or services to users is not a vi-

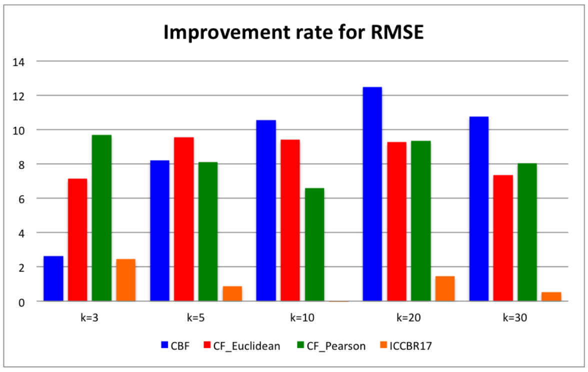


Fig. 13. Improvement rate for RMSE based on MovieLens 1m

able business strategy anymore. This might be the case if the users might not want to select popular products all the time but want to broad their taste. By offering a recommendation method that provides more accurate recommendations of items found in the long tail a solution can be provided to users that we want to be loyal to a business by providing an effective recommender system that includes items from the long tail as well or in a total different scenario this could be recommending news content to users that it's difficult for them to find elsewhere.

A typical business model might usually include different types of recommendations such as (a) popular items (b) what users with similar purchase history like (c) content-based (d) long tail and many other possible combinations as well. In this paper we have concentrated on the development of a recommendation method that improves the accuracy of the recommendations in the long tail, which we consider to be an important business model to the recommendation process. Our proposed recommendation method fits this business model as it provides more accurate recommendation to users or to explain this more accurately possible customers of e-Commerce or other relevant websites such as social networks. The proposed method has been evaluated using two real datasets with the results validating it in most scenarios and it has been compared against a number of alternative methods used as baselines. Moreover, the well known error rating prediction metrics MAE and RMSE have been used along with an overall improvement rate for each of the metrics and datasets.

The proposed recommendation method can assist further the business model of websites by increasing the user experience. This is important since for increasing sales we need happy users that like a service and will come back to use it again. Furthermore, the search burden of users will be low since more relevant products or services will be available without searching for them and the processing power required by a vendor will be reduced, resulting in a win-win situation for both the customers and the vendors. Additionally to the aforementioned, a good quality recommendation method for the long tail can become a good social opportunity as well where a diversity of products, manufacturers and contents is good for developing a marketplace where is possible for a diverse set of products to find potential customers. Recommending in the long tail is a way to help avoiding recommendations from a well known list of products only.

In addition, our switching method can be adopted to solve the cold-start problem through a switching when a new user or new item is added to the system. However, in order to calculate the similarity a knowledge based system is required in this case to obtain information about the new user or items

## 6. Conclusions and Future Work

In this paper we presented a novel switching method that utilizes a collaborative filtering method based on multi-level similarity features and content based filtering using user history. The proposed method has been experimentally evaluated using two real datasets and most known and established rating prediction accuracy metrics with the results being improved when compared to alternative. In the experiments it is shown that the proposed approach outperforms all alternatives in most situations. The proposed method outperforms the CF, CBF, the multi-level CF and ICCBR17 approach in most evaluation tests. since the long tail recommendation problem occurred when there are no sufficient data to build a reliable predictive model in the tail, our switching method that adopts the two popular techniques, is a reasonable solution. It can provide similar items for those less popular and fail in one technique. The method presented in this paper does not need to pre-process the data before executing the recommendation. Furthermore, this method does not need to save more information, because in both techniques we use the same information (the users rating histories). We explained that our proposed method improves the accuracy for items in the long tail. Furthermore, based on the comparisons with the traditional CF similarity measures and state-of-the-art alternatives, it is shown that the proposed method is robust against the long tail recommendation problem. In the future, we aim to evaluate the proposed method using an online platform to allow the observation of its effectiveness and the measurement of the quality of the recommendations from a real user perspective.

## References


[1] Gharbi Alshammari, Jose L Jorro-Aragoneses, Nikolaos Polatidis, Stelios Kapetanakis, and Miltos Petridis. A switching approach that improves prediction accuracy for long tail recommendations. *Intelligent System conference*, 2019. (in press).

[2] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[3] Fatih Gedikli and Dietmar Jannach. Recommending based on rating frequencies: Accurate enough? In *Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at UMAP'10 (ITWP'10)*, pages 65–70, 2010.

[4] Buhwan Jeong, Jaewook Lee, and Hyunbo Cho. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Information Sciences*, 180(5):602–612, 2010.

[5] Joseph A Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22(1-2):101–123, 2012.

[6] Yue Shi, Martha Larson, and Alan Hanjalic. 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.

[7] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[8] Chris Anderson. The Long Tail: Why the Future of Business Is Selling Less of More by Chris Anderson. *Journal of Product Innovation Management*, 24(3):1–30, 2007.

[9] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Value-aware item weighting for long-tail recommendation. *arXiv preprint arXiv:1802.05382*, 2018.

[10] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[11] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[12] Nikolaos Polatidis and Christos K Georgiadis. A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications*, 48:100–110, 2016.

[13] Mubbashir Ayub, Mustansar Ali Ghazanfar, Muazzam Maqsood, and Asjad Saleem. A jaccard base similarity measure to improve performance of cf based recommender systems. In *2018 International Conference on Information Networking (ICOIN)*, pages 1–6. IEEE, 2018.

[14] Kaili Shen, Yun Liu, and Zhenjiang Zhang. Modified similarity algorithm for collaborative filtering. In *International Conference on Knowledge Management in Organizations*, pages 378–385. Springer, 2017.

[15] Gharbi Alshammari, Stelios Kapetanakis, Nikolaos Polatidis, and Miltos Petridis. A triangle multi-level item-based collaborative filtering method that improves recommendations. In *International Conference on Engineering Applications of Neural Networks*, pages 145–157. Springer, 2018.

[16] Gharbi Alshammari, Jose L Jorro-Aragoneses, Stelios Kapetanakis, Miltos Petridis, Juan A Recio-García, and Belén Díaz-Agudo. A hybrid cbr approach for the long tail problem in recommender systems. In *International Conference on Case-Based Reasoning*, pages 35–45. Springer, 2017.

[17] Rahul Katarya and Om Prakash Verma. Effectual recommendations using artificial algae algorithm and fuzzy c-mean. *Swarm and Evolutionary Computation*, 36:52–61, 2017.

[18] Shouxian Wei, Xiaolin Zheng, Deren Chen, and Chaochao Chen. Electronic Commerce Research and Applications A hybrid approach for movie recommendation via tags and ratings q. 18:83–94, 2016.

[19] Zhenhua Tan and Liangliang He. An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. *IEEE Access*, 5:27211–27228, 2017.

[20] Shuang-Bo Sun, Zhi-Heng Zhang, Xin-Ling Dong, Heng-Ru Zhang, Tong-Jun Li, Lin Zhang, and Fan Min. Integrating triangle and jaccard similarities for recommendation. *PloS one*, 12(8):e0183570, 2017.

[21] Vladislav Grozin and Alla Levina. Similar product clustering for long-tail cross-sell recommendations.

[22] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9):896–907, 2012.

[23] Yoon Joo Park. The adaptive clustering method for the long tail problem of recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1904–1915, aug 2013.

[24] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18. ACM, 2008.

[25] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, page 39, 2010.

[26] Susan Craw, Ben Horsburgh, and Stewart Massie. Music recommendation: Audio neighbourhoods to discover music in the long tail. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9343:73–87, 2015.

[27] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[28] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.