An Improved Collaborative Movie Recommendation System using Computational Intelligence

Zan Wang
Department of Software Engineering,
School of Computer Software,
Tianjin University
Tianjin, 300072, P.R. China
wangzan@tju.edu.cn

Xue Yu*, Nan Feng
Department of Information
Management & Management Science,
College of Management and
Economics, Tianjin University
Tianjin, 300072, P.R. China
{yuki, fengnan}@tju.edu.cn

Zhenhua Wang
American Electric Power
Gahanna, OH 43230, United States
zhw.powersystem@gmail.com

Abstract—**Recommendation systems have become prevalent in recent years as they dealing with the information overload problem by suggesting users the most relevant products from a massive amount of data. For media product, online collaborative movie recommendations make attempts to assist users to access their preferred movies by capturing precisely similar neighbors among users or movies from their historical common ratings. However, due to the data sparsely, neighbor selecting is getting more difficult with the fast increasing of movies and users. In this paper, a hybrid model-based movie recommendation system which utilizes the improved K-means clustering coupled with genetic algorithms (GA) to partition transformed user space is proposed. It employs principal component analysis (PCA) data reduction technique to dense the movie population space which could reduce the computation complexity in intelligent movie recommendation as well. The experiment results on Movielens dataset indicate that the proposed approach can provide high performance in terms of accuracy, and generate more reliable and personalized movie recommendations when compared with the existing methods.**

Keywords—Movie recommendation, Collaborative filtering, Sparsity data, Genetic algorithms, K-means

I. INTRODUCTION

Fast development of internet technology has resulted in explosive growth of available information over the last decade. Recommendation systems (RS), as one of the most successful information filtering applications, have become an efficient way to solve the information overload problem. The aim of Recommendation systems is to automatically generate suggested items (movies, books, news, music, CDs, DVDs, webpages) for users according to their historical preferences and save their searching time online by exacting useful data.

Movie recommendation is the most widely used application coupled with online multimedia platforms which aims to help customers to access preferred movies intelligently from a huge movie library. A lot of work has been done both in the academic and industry area in developing new movie recommendation algorithms and extensions. The majority of existing recommendation systems is based on collaborative filtering (CF) mechanism [1-3] which has been successfully developed in the past few years. It first collects ratings of movies given by individuals and then recommends promising movies to target customer based on the "like-minded" individuals with similar tastes and preferences in the past. There have been many famous online multimedia platforms (e.g., youtube.com, Netflix.com, and douban.com) incorporated with CF technique to suggest media products to their customers. However, traditional recommendation systems always suffer from some inherent limitations: poor scalability, data sparsity and cold start problems [3, 4]. A number of works have developed model-based approaches to deal with these problems and proved the benefits on prediction accuracy in RS [5-8].

Model-based CF uses the user-item ratings to learn a model which is then used to generate online prediction. Clustering and dimensionality reduction techniques are often employed in model-based approaches to address the data sparse problem [5, 8-9]. The sparsity issues arise due to the insufficiency of user's history rating data and it is made even more severe in terms of the dramatically growth of users and items. Moreover, high-dimensional rating data may cause it difficult to extract common interesting users by similarity computation, which results in poor recommendations. In the literature, there have been many model-based recommendation systems developed by partitioning algorithms coupled, such as K-means and self-organizing maps (SOM) [15-18, 20]. The aim of clustering is to divide users into different groups to form "like-minded" (nearest) neighbors instead of searching the whole user space, which could dramatically improve the system scalability. It has been proved that clustering-based recommendation systems outperform the pure CF-based ones in terms of efficiency and prediction quality [7, 9-11]. In many works, the clustering methods are conducted with the entire dimensions of data which might lead to somewhat

*Corresponding author. Tel.: +86 22 27406125; Fax: +86 22 87401540 inaccuracy and consume more computation time. In general, making high quality movie recommendations is still a challenge, and exploring an appropriate and efficiency clustering method is a crucial problem in this situation.

To address challenges aforementioned, a hybrid model-based movie recommendation approach is proposed to alleviate the issues of both high dimensionality and data sparsity. In this article, we construct an optimized clustering algorithm to partition user profiles which have been represented by denser profile vectors after Principal Component Analysis transforming. The whole system consists of two phases, an online phase, and an offline phase. In offline phase, a clustering model is trained in a relatively low dimensional space, and prepares to target active users into different clusters. In online phase, a TOP-*N* movie recommendation list is presented for an active user due to predicted ratings of movies. Furthermore, a genetic algorithm (GA) is employed in our new approach to improve the performance of *K*-means clustering, and the improved clustering algorithm is named as GA-KM. We further investigate the performance of the proposed approach in Movieslens dataset. In terms of accuracy and precision, the experiment results prove that the proposed approach is capable of providing more reliable movie recommendations comparing with the existing cluster-based CF methods.

The remainder of this paper is organized as follows: section 2 gives a brief overview on collaborative recommendation systems and clustering-based collaborative recommendation. Then we discuss the development of our proposed approach called PCA-GAKM movie recommendation system in detail in Section 3. In section 4, experiment results on movielens dataset and discussion are described. Finally, we summarize this paper and the future work is given.

II. Related Work

A. Movie Recommendation Systems based on Collaborative Filtering

Recommendation systems (RS), introduced by Tapestry project in 1992, is one of the most successful information management systems [12]. The practical recommender applications help users to filter mass useless information for dealing with the information overloading and providing personalized suggestions. There has been a great success in e-commerce to make the customer access the preferred products, and improve the business profit. In addition, to enhance the ability of personalization, recommendation system is also widely deployed in many multimedia websites for targeting media products to particular customers. Nowadays, Collaborative filtering (CF) is the most effective technique employed by movie recommendation systems, which is on the basis of the nearest-neighbor mechanism. It is on the assumption that people who have similar history rating pattern may be on the maximum likelihood that have the same preference in the future. All "like-minded" users, called neighbors, are derived from their rating database that is recording evaluation values to movies. The prediction of a missing rating given by a target user can be inferred by the weighted similarity of his/her neighborhood.

Reference [6] divides CF techniques into two important classes of recommender systems: memory-based CF and model-based CF. Memory-based CF operate on the entire user space to search nearest neighbors for an active user, and automatically produce a list of suggested movies to recommend. This method suffers from the computation complexity and data sparsity problem. In order to address computational and memory bottleneck issues, Sarwar et al. proposed an item-based CF in which the correlations between items are computed to form the neighborhood for a target item [4]. In their empirical studies, it is proved that item-based approach can shorten computation time apparently while providing comparable prediction accuracy.

Model-based CF, on the other hand, develops a pre-build model to store rating patterns based on user-rating database which can deal with the scalability and sparsity issues. In terms of recommendation quality, model-based CF applications can perform as well as memory-based ones. However, model-based approaches are time-consuming in building and training the offline model which is hard to be updated as well. Algorithms that often used in model-based CF applications include Bayesian networks [6], clustering algorithms [9-11], neural networks [13], and SVD (Singular Value Decomposition) [5, 14]. While traditional collaborative recommendation systems have their instinct limitations, such as computational scalability, data sparsity and cold-start, and these issues are still challenges that affect the prediction quality. Over the last decade, there have been high interests toward RS area due to the possible improvement in performance and problems solving capability.

B. Clustering-based Collaborative Recommendation

In movie recommendation, clustering is a widely used approach to alleviate the scalability problem and provides a comparable accuracy. Many works have proved with experiments that the benefits of clustering-based CF frameworks [15-18]. The aim of clustering algorithms is to partition objects into clusters that minimize the distance between objects within a same cluster to identify similar objects. As one of model-based CF methods, clustering-based CF is used to improve *k*-nearest neighbor (*k*-NN) performance by prebuilding an offline clustering model. Typically, numbers of users can be grouped into different clusters based on their rating similarity to find "like-minded" neighbors by using the clustering technique. Then the clustering process is performed offline to build the model. When a target user arrived, the online module assigns a cluster with a largest similarity weight to him/her, and the prediction rating of a specified item is computed based on the same cluster numbers instead of searching whole user space.

According to early studies in [3, 6], CF coupled with clustering algorithms is a promising schema to provide accuracy personal recommendations and address the large scale problems. But they also concluded that good performance of clustering-based CF depends on appropriate clustering techniques and the nature of dataset as well. Li and Kim applied fuzzy *K*-means clustering method to group items which combined the content information for similarity measurement to enhance the recommendation accuracy [9]. In the conclusion of their work, it shows that the proposed cluster-based approach is capable of dealing with the cold start problem. Furthermore, Wang et al. developed [19] a new approach to cluster both the rows and columns fuzzily in order to condense the original user rating matrix. In Kim and Ahn's research, a new optimal *K*-means clustering approach with genetic algorithms is introduced to make online shopping market segmentation [10]. The proposed approach is tested to exhibit better quality than other widely used clustering methods such as pure *K*-means and SOM algorithms in the domain of market segmentation, and could be a promising tool for e-commerce recommendation systems.

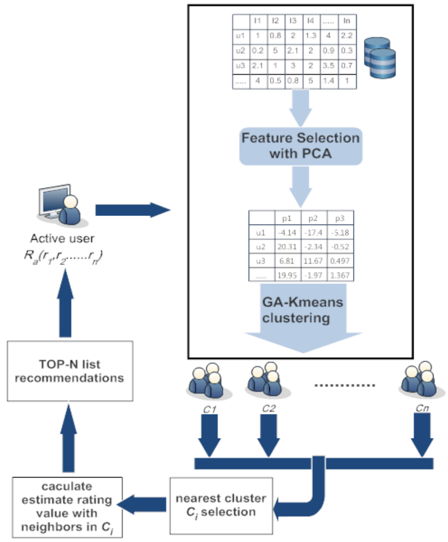
Liu and Shih proposed two hybrid methods that exploited the merits of the Weighted RFM-based and the preference-based CF methods to improve the quality of recommendations [20]. Moreover, *K*-means clustering is employed to group customers based on their enhanced profile. The experiments prove that the combined models perform better than the classical *K*-NN mechanism. Xue et al. proposed a novel CF framework that uses clustering technique to address data sparsity and scalability in common CF [7]. In their work, *K*-means algorithm is employed to classify users for smoothing the rating matrix that is to generate estimated values for missing ratings corresponding to cluster members. In latter recommendation phrase, the clustering result is utilized to neighborhood selection for an active user. The experiment results show that the novel approach can demonstrate significant improvement in prediction accuracy. Georgiou and Tsapatsoulis developed a genetic algorithm based clustering method which allows overlapping clusters to personalized recommendation, and their experiment findings show that the new approach outperforms *K*-means clustering in terms of efficiency and accuracy [21].

The above works have proved that clustering-based CF systems show more accuracy prediction and help deal with scalability and data sparse issues.

III. PCA-GAKM BASED COLLABORATIVE FILTERING FRAMEWORK

In this section, we aim at developing a hybrid cluster-based model to improve movie prediction accuracy, in which offline and online modules are coupled to make intelligent movie recommendations. Traditional CF search the whole space to locate the k-nearest neighbors for a target user, however, considering the super high dimensionality of user profile vectors, it is hard to calculate a similarity to find like-minded neighbors based on ratings which leads to poor recommendation because of sparse. To address such an issue, our offline clustering module involves two phases: 1) to concentrate feature information into a relatively low and dense space using PCA technique; 2) To build an effective GA-KM clustering algorithm based on the transformed user space.

Fig 1 shows an overview of the new approach: offline module represented by light flow arrows, is used to optimize and train the user profiles into different clusters on the basis of history rating data; online module is real-time movie recommendation noted with dark flow arrows, to which a target user's rating vector input, and come out with a TOP-N movie recommendation list. We explain the details in the following.



	I1	I2	I3	I4	.....	In
u1	1	0.8	2	1.3	4	2.2
u2	0.2	5	2.1	2	0.9	0.3
u3	2.1	1	3	2	3.5	0.7
.....	4	0.5	0.8	5	1.4	1

	p1	p2	p3
u1	-4.14	-17.4	-5.18
u2	20.31	-2.34	-0.52
u3	6.81	11.67	0.497
.....	19.95	-1.97	1.367

Feature Selection with PCA
GA-Kmeans clustering
Active user
$R_a(r_1,r_2,\ldots\ldots r_n)$
TOP-N list recommendations
C1
C2
....
Cn
caculate estimate rating value with neighbors in C_i
nearest cluster C_i selection

Fig 1. Overview of proposed movie recommendation system framework

A. *Pre-processing Data using PCA*

In this section, we employ a linear feature extraction technique to transfer the original high space into a relatively low space in which carries denser feature information. Since the high dimensionality of user-rating matrix which is mostly empty at the beginning makes the similarity computation very difficult, our approach is started with PCA-based dimension reduction process. As one of the most successful feature extraction techniques, PCA is widely used in data prefilling and dimensional reduction of collaborative filtering systems [14][22-23].

The main idea of PCA is to convert the original data to a new coordinate space which is represented by principal component of data with highest eigenvalue. The first principal component vector carries the most significant information after ordering them by eigenvalues from high to low. In general, the components of lesser significance are ignored to form a space with fewer dimensions than the original one. Suppose we have user-rating $m \times n$ matrix in which n-dimension vector represents user's profile. It turns out the n principal components after performing eigenvalue decomposition, and we select the only first d components ($d \ll n$) to keep in the new data space which is based on the value of accumulated proportion of 90% of the original one. As a result, the reduced feature vectors from PCA are prepared to feed to GA-KM algorithm for classification.

B. An Enhanced K-means Clustering Optimized by Genetic Algorithms

Memory-based CF systems suffer from two main common flaws: cold-start and data sparse. Many research works have proved benefits of cluster-based CF in terms of the increased quality of recommendation and robustness. The objective of this section is to propose an effective classification method to ensure the users who have the same preference could fall into one cluster to generate accurate like-minded neighbors. The GA-KM algorithm we employed in this work can be roughly performed in two phases:

- *K*-means clustering

K-means algorithm is one of the most commonly used clustering approaches due to its simplicity, flexibility and computation efficiency especially considering large amounts of data. *K*-means iteratively computes *k* cluster centers to assign objects into the most nearest cluster based on distance measure. When center points have no more change, the clustering algorithm comes to a convergence. However, *K*-means lacks the ability of selecting appropriate initial seed and may lead to inaccuracy of classification. Randomly selecting initial seed could result in a local optimal solution that is quite inferior to the global optimum. In other words, different initial seeds running on the same dataset may produce different partition results.

Given a set of objects $(x_1, x_2, \cdots x_n)$, where each object is an m-dimensional vector, *K*-means algorithm aims to automatically partition these objects into *k* groups. Typically, the procedure consists of the following steps [24-25]:

1) choose *k* initial cluster centers $C_j, j=1,2,3\ldots k;$
2) each x_i is assigned to its closest cluster center according to the distance metric;
3) compute the sum of squared distances from all members in one cluster:

$$J = \Sigma_{j=1}^{k} \Sigma_{i \in C_{temp}} \left\| x_i - M_j \right\|^2 \qquad (1)$$

where M_j denotes the mean of data points in C_{temp};

4) if there is no further change, then the algorithm has converged and clustering task is end; otherwise, recalculate the M_j of *k* clusters as the new cluster centers and go to step2.

To overcome the above limitations, we introduce genetic algorithm to merge with *K*-means clustering process for the enhancement of classification quality around a specified *k*.

- Genetic algorithms

Genetic algorithms (GA) are inspired by nature evolutionary theory which is known for its global adaptive and robust search capability to capture good solutions [26]. It can solve diverse optimization problems with efficiency due to its stochastic search on large and complicated spaces. The whole process of GA is guided by Darwin's nature survival principle and provides a mechanism to model biological evolution. A GA utilizes a population of "individuals" as chromosomes, representing possible solutions for a given problem. Each chromosome contains number of genes which is used to compute fitness to determine the likelihood of reproduction for the next generation. Typically, a chromosome with the fittest value will be more likely to reproduce than unfit ones. GAs iteratively creates new populations replace of the old ones by selecting solutions (chromosomes) on the basis of pre-specified fitness function. During each successive iteration, three genetic operators are executed to construct the new generation known as selection, crossover and mutation. Selection process selects a proportion of the current population to breed a new generation according to their fitness value. Crossover operator allows swapping a portion of two parent chromosomes for each other to be recombined into new offspring. Mutation operator randomly alters the value of a gene to produce offspring. All above operators provide the means to extend the diversity of population over time and bring new information to it. Finally, the iterations tend to terminate when the fitness threshold is met or a pre-defined number of generations is reached.

A common drawback of *K*-means algorithm has described above that sensitivity selection of initial seeds could influence final output and easy to fall into local optimum. In order to avoid the premature convergence of *K*-means clustering, we considered a genetic algorithm as the optimization tool for evolving initial seeds in the first step of *K*-means process in order to identify optimal partitions. In our study, a chromosome with *k* genes is designed for *k* cluster centers as $(x_1, x_2, \cdots x_k)$, where x_i is a vector with *n* dimensions. During the evolution process, we applied fitness function to evaluate the quality of solutions that is:

$$f(chromosome) = \sum_{x_j \in X} \min_{1 \le i \le k} (dist(C_i, x_j)) \qquad (2)$$

The fitness value is the sum of distances for all inner points to their cluster centers and tries to minimize the values which correspond to optimized partitions. In every successive iteration, three genetic operators precede to construct new populations as offspring according to the fittest survival principles. The populations tend to converge to an optimum chromosome (solution) when the fitness criterion is satisfied. Once the optimal cluster centers have come out, we use them as initial seeds to perform *K*-means algorithm in the last step of clustering. Pseudo-code of the hybrid GA-KM approach is presented as follows, and other configuration parameters will be pointed in Fig 2.

```
Algorithm：GA-KM Pseudo-code
Initialization:
        Parameters Initialization:
                Maximum Iterations $Tmax$ = 200 and iteration $t$=0;
                Population Size $popnum$ = 50;
                Clusters Number: $k$;
                Probability of Crossover: $Pc$;
                Probability of Mutation: $Pm$;
                Fitness function：minimize the total distance of every sample to its nearest center
                        $f(chromosome) = \sum_{x_j \in X} \min_{1 \le i \le k} (dist(C_i, x_j))$
        Population Initialization:
                Generate the initial population $P(0)$ randomly, each individual consists of $k$ centers;
While ( $t < Tmax$ )
{
        For $i$ = 1 to $popnum$
                $fi = f(chromosomei)$;
        End
        Optimization Reserved for Each Population;
        Selection Operator:
                Roulette Selection;
        Crossover Operator:
                Select $\alpha \in [1,k]$ randomly;
                Crossover parent chromosomes with probability Pc:
                        Parent chromosomes $C_{i1}$,$C_{i2}$, new pair of Children chromosomes $C_{i1}'$,$C_{i2}'$:
                                $C_{i1}' = [C_{i1}(1{:}\ \alpha, :); C_{i2}(\alpha+1{:}\ end, :)]$
                                        $C_{i2}' = [C_{i2}(1{:}\ \alpha, :); C_{i1}(\alpha+1{:}\ end, :)]$

              Compare $C_{i1}'$,$C_{i2}'$ with $C_{i1}$,$C_{i2}$, if the children chromosomes are better than the parents, the parents will
              be replaced with the children;
                Replace the worst two chromosomes in population with $C_{i1}'$,$C_{i2}'$;
        Mutation Operator:
                Mutate each center of the best chromosomes with probability Pm respectively:
                        For each center $C_i$:
                                Replace $C_i$ with random center in all samples;
                Compare $C_{new}$ with $C_i$, if it is better, $C_i$ will be replaced with $C_{new}$ ;
        $t = t + 1$;
}
Get the initial $k$ centers with the optimal fitness value.
$K$-means Optimization:
        Generate new clusters and new $k$ centers;
```

Fig 2. GA-KM clustering algorithm procedure

The offline model was constructed by our PCA-GAKM approach, and once the target user is reached, we calculate the most interesting movies as TOP-N recommendation list online from a cluster neighborhood instead of searching the whole user space. The estimated rating value for an un-rated movie given by U_a is predicted as follows [27]:

$$P_{Ua,item} = \overline{R_u} + \frac{\sum_{y \in C_x} sim(U_a, y) \times (R_{y,i} - \overline{R_y})}{\sum_{y \in C_x} (|sim(U_a, y)|)} \qquad (3)$$

where $\overline{R_u}$ is average rating score given by U_a, C_x is a set of neighbors belonging to one common cluster with

U_a, $\overline{R_y}$ denotes the average rating given by U_a's neighbor y, $sim(U_a\ , y)$ is a similarity function based on Pearson correlation measure to decide the similarity degree between two users.

IV. Experiments and Results

In this section, we describe the experimental design, and empirically investigate the proposed movie recommendation algorithm via PCA-GAKM technique and compare its performance with benchmark clustering-based CF. Finally the results will be analyzed and discussed. We carried out all our experiments on Dual Xeon 3.0GHz, 8.0GB RAM computer and run Matlab R2011b to simulate the model.

A. Data Set and Evaluation Criteria

We consider the well-known Movielens dataset to conduct the experiments, which is available online, including 100,000 ratings by 943 users on 1,682 movies, and assigned to a discrete scale of 1-5. Each user has rated at least 20 movies. We use φ to describe the sparsity level of dataset: φ_{ml}=1-100,000/943×1682=0.9369. Then the dataset was randomly split into training and test data respectively with a ratio of 80%/20%. We utilized training data to build the offline model, and the remaining data were used to make prediction. To verify the quality of recommendation, we employed the mean absolute error (MAE), precision, recall as evaluation measures which have been widely used to compare and measure the performance of recommendation systems. The MAE is a statistical accuracy metric which measures the average absolute difference between the predicted ratings and actual ratings on test users as shown in Eq.(4). A lower MAE value corresponds to more accurate predictions.

$$MAE = \frac{\sum \left| \tilde{P}_{i,j} - r_{i,j} \right|}{M} \tag{4}$$

where M is the total number of predicted movies, $\tilde{P}_{i,j}$, represents the predicted value for user i on item j, and $r_{i,j}$ is the true rating.

To understand whether users are interested with the recommendation movies, we employ the precision and recall metrics which are widely used in movie recommender systems to evaluate intelligence level of recommendations. Precision is the ratio of interesting movies retrieved by a recommendation method to the number of recommendations. Recall gives the ratio of interesting movies retrieved that is considered interesting. These two measures are clearly conflict in nature because increasing the size of recommended movies N leads to an increase in recall but decrease the precision. The precision and recall for *Top-N* (N is the number of predicted movies) recommendation are defined in (5) & (6) respectively.

$$precision = \frac{\left| interesting \bigcap TopN \right|}{N} \tag{5}$$

$$recall = \frac{\left| interesting \bigcap TopN \right|}{\left| interesting \right|} \tag{6}$$

B. Experimental Designs

We try to conduct different clustering algorithms – K-means, SOM, and the proposed GA-KM on a relatively low dimension space after PCA transformation. K-means is easy with efficiency, but sensitive for initial cluster and often convergence to a local optimum. SOM, as an artificial neural network, has been applied to many intelligent systems for its good performance. In our GA optimal processing, we use Euclidean distance measure to decide the similarity of n-dimensional vectors in search space. The initial population is generated with the size of 50 and number of generation: *Tmax*=200. Parameter N represents the number of movies on the recommendation list. To decide the cluster number K suitably, we first make a robust estimation on unrated scores by performing global K-means clustering operations on dataset where K varies from 4 to 28. It has been seen in Fig 3 that smaller MAE values occurs when K is between 12 and 18. Thus we set K to 16 as the total number of clusters to guide our numerical experiments.

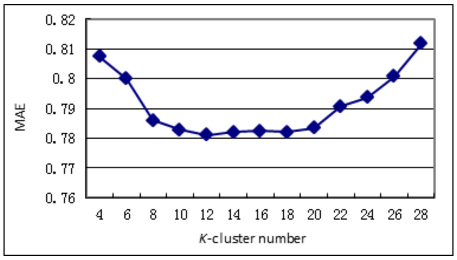


Fig 3. Precision error on different cluster numbers

In this part, we use an all-but-10 method, in which we randomly held out ten ratings for each user in test data that should be predicted based on model. Prediction is calculated for the hidden votes by three clustering algorithms to compare the recommendation accuracy. In addition, to examine the prediction precision, we employed UPCC (Pearson Correlation Coefficient based CF) and up-to-date clustering-based CF methods to compare with the proposed hybrid clustering approach.

We also conduct experiments to verify quality of the new hybrid model. Here cold-start users are defined as users who have rated less than 5 movies. We get five ratings visible for each test user; the rest of ratings replaced with null, and tried to prediction their values. Given5 data is designed to test cold-start problem caused by a new user with little history information. To normal target users, we also build Given10 to Given20 and Allbut5 to Allbut10 test data to generate recommendation. In each above experiment, we repeat five times for randomly training and test datasets, and average the results.

C. Results and Discussion

The sparse of user-item rating matrix makes it hard to find real neighbors to form the final recommendation list. In our experiments, we compare the performances and some trends of the existing baseline CF movie recommendation systems with our approach, while the neighborhood size varies from 5-60 in an increment of 5. Detail explanation is showed as the follows from experiment results:

1) Performance of PCA-GAKM CF approach

We first try to evaluate the movie recommendation quality with the traditional cluster-based CFs. Fig 4 shows that all methods tries to reach the optimum prediction values where the neighborhood size varies from 15-20, and it becomes relatively stable around 60 nearest neighbors. All clustering with PCA algorithms performed better accuracy than pure cluster-based CFs. We consider that PCA process could be necessary to dense the original user-rating space, and then improve the partition results. Without the first step of dimensional reduction, GAKM and SOM gave very close MAE values and it seems that GAKM produce slightly better prediction than SOM. When coupling with PCA technique, GAKM shows a distinct improvement on recommendation accuracy compared with SOM. Moreover, the proposed PCA-GAKM performs apparently high accuracy among all the algorithms, and produces the smallest MAE values continually where the neighbor size varies. All *K*-means clustering CF generate increasing MAE values which indicate the decreasing quality for recommendation due to sensitiveness of the algorithm. Traditional user-based CF produces relatively worse prediction compared with the basic clustering-based methods.

To exam the difference of predictive accuracy between our proposed method and other comparative cluster-based methods, we applied *t*-test in the recommendation results. As shown in Table 1, the differences between MAE values are statistically significant at the 1% level. Therefore, we can affirm that the proposed PCA-GAKM outperforms with respect to the comparable cluster-based methods.

TABLE I. THE *T*-TEST RESULTS FOR VARIOUS CLUSTER-BASED METHODS IN TERMS OF MAEs.

			PCA-GAKM	
Method	**Mean**	**Std. dev**	***t*-Value**	**Sig***
PCA-SOM	0.8018	5.900e-03	9.0633	.000
SOM-CLUSTER	0.8078	2.589e-03	16.5194	.000
UPCC	0.8232	1.292e-03	28.9869	.000
KMEANS-CLUSTER	0.8175	2.244e-03	23.3691	.000
PCA-KMEANS	0.8412	2.845e-03	37.0515	.000
GAKM-CLUSTER	0.8040	2.786e-03	13.8541	.000
PCA-GAKM	0.7821	4.747e-03		

*Statistically significant at the 1% level.

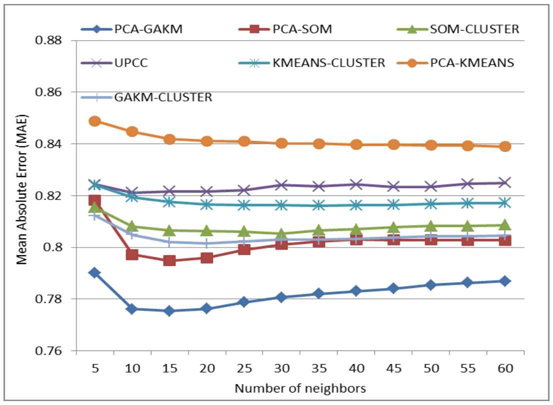


Fig 4. Comparing accuracy with existing clustering-based CF

2) Precision of PCA-GAKM CF approach

To analyze precision of recommendation, we fix the neighbor size *n*=20. As seen from Fig 5, the overall precisions improve with the increasing number of recommendation, and the PCA-GAKM generates higher precision rates which indicate that it can recommend more interesting and reliable movies to users than other clustering-based algorithms when a relatively small number of movies on recommendation list are considered. In addition, Fig 6 compares the recall rates of user interesting movies, and it's apparently that PCA-GAKM still provide greater recall rates with each value in *N* (the number of recommendation). The existing cluster-based CFs show lower precision and recall rates comparing to our optimal clustering approach.

3) Recommendation of cold-start users

We finally experiment the cold-start problem with "less information" users who has rated few movies in history. It is understandable that searching neighbors in high dimensional space become difficult for cold-start users with few ratings. Fig 7 enables us to discover that clustering coupled with PCA methods may produce a generalized improvement in prediction accuracy for cold-start users. Among examined clustering methods, the proposed PCA-GAKM seems to have the best performance in alleviating cold-start with the satisfactory MAE values. With increasing number of ratings used to make prediction, all approaches show the similar trend that prediction accuracy is getting higher as presented in Fig 7.

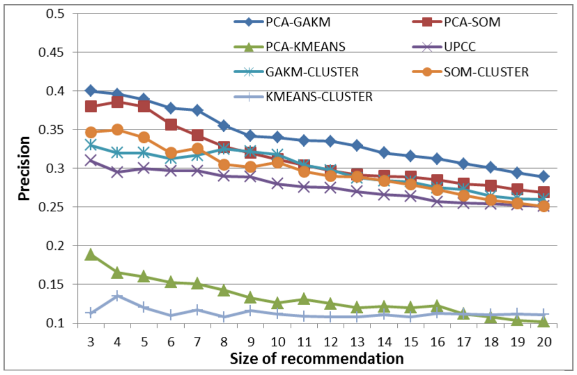


Fig 5. Precision comparison with existing cluster-based CF

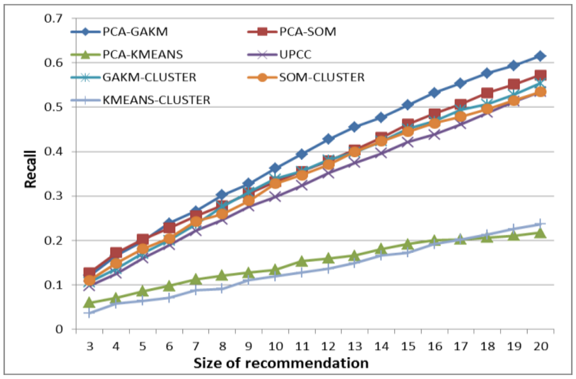


Fig 6. Recall comparison as the recommendation size grows

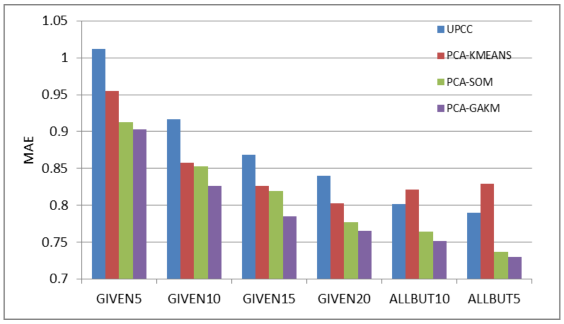


Fig 7. MAE comparisons in different rating reveal level

V. Conclusions and Future Work

In this paper we develop a hybrid model-based CF approach to generate movie recommendations which combines dimensional reduction technique with clustering algorithm. In the sparse data environment, selection of "like-minded" neighborhood on the basis of common ratings is a vital function to generate high quality movie recommendations. In our proposed approach, feature selection based on PCA was first performed on whole data space, and then the clusters were generated from relatively low dimension vector space transformed by the first step. In this way, the original user space becomes much denser and reliable, and used for neighborhood selection instead of searching in the whole user space. In addition, to result in best neighborhood, we apply genetic algorithms to optimize *K*-means process to cluster similar users. Based on the Movielens dataset, the experimental evaluation of the proposed approach proved that it is capable of providing high prediction accuracy and more reliable movie recommendations for users' preference comparing to the existing clustering-based CFs. As for cold-start issue, the experiment also demonstrated that our proposed approach is capable of generating effective estimation of movie ratings for new users via traditional movie recommendation systems.

As for future work, we will continue to improve our approach to deal with higher dimensionality and sparsity issues in practical environment, and will explore more effective data reduction algorithms to couple with clustering-based CF. Furthermore, we will study how the variation number of clusters may influence the movie recommendation scalability and reliability. To generate high personalized movie recommendations, other features of users, such as tags, context, and web of trust should be considered in our future studies.

Acknowledgements

The research was supported by the National Natural Science Foundation of China (Nos. 71271149, 70901054 and 61202030). It was also supported by the Program for New Century Excellent Talents in University (NCET). The authors are very grateful to all anonymous reviewers whose invaluable comments and suggestions substantially helped improve the quality of the paper.

References

[1] G. Adomavicius, A. Tuzhilin, "Toward the next generation of recommender system: A survey of the state-of-the-art and possible extensions," IEEE Trans on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.

[2] G. Linden, B. Smith and J. York, "Amazon.com recommendations: Item to item collaborative filtering," IEEE Internet Computing, vol.7, no.1, pp. 76–80, 2003.

[3] B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in Proceedings of the international conference on computer and information technology, 2002.

[4] B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithm," in Proceedings of the 10th International World Wide Web Conference. Hong Kong, 2001, pp. 285–295.

[5] B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Application of Dimensionality Reduction in Recommender System-A Case Study," in ACM 2000 KDD Workshop on Web Mining for e-commerce-Challenges and Opportunities, 2000.

[6] J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of Predictive Algorithms for collaborative filtering," in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.

[7] G. Xue, C. Lin, Q. Yang, et al, "Scalable Collaborative Filtering Using Cluster-based Smoothing," in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Brazil: ACM Press, 2005, pp. 114–121.

[8] F. Gao, C. Xing and Y. Zhao, "An effective algorithm for dimensional reduction in collaborative filtering," in Lecture Notes on Computer Science, Springer Berlin, 2007, pp. 75–84.

[9] Q. Li and B. M. Kim, "Clustering approach for hybrid recommendation system," in Proceedings of the International Conference on Web Intelligence, 2003, pp. 33–38.

[10] K. Kim and H. Ahn, "A recommender system using GA K-means clustering in an online shopping market," Expert Systems with Application, vol. 34, no. 2, pp. 1200–1209, 2008.

[11] A. Kohrs and B. Merialdo, "Clustering for collaborative Filtering Applications," in Proceedings of CIMCA'99, Vienna: IOS Press, 1999, pp. 199–204.

[12] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[13] D. Billsus and M. J. Pazzani, "Learning collaborative information filters," in Proceedings of the 15th International Conference on Machine Learning, Madision, 1998, pp. 46–53.

[14] K. Goldberg, T. Roeder, D. Gupta and C.Perkins, "Eigentaste: A constant time collaborative filtering algorithm," Information Retrieval, vol. 4, no. 2, pp. 133–151, 2001.

[15] K. Q. Truong, F. Ishikawa and S. Honiden, S, "Improving accuracy of recommender system by item clustering," Transactions on Information and Systems, E90-D(9), pp. 1363–1373, 2007.

[16] G. Pitsilis, X. L. Zhang and W. Wang, "Clustering recommenders in collaborative filtering using explicit trust information," in Proceedings of the 5th IFIP WG 11.11 International Conference on Trust Management, vol.358, 2011, pp. 82–97.

[17] M. Zhang and N. Hurley, "Novel item recommendation by user profile partitioning," in Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, 2009, pp. 508–515.

[18] C. Huang and J. Yin, J, "Effective association clusters filtering to cold-start recommendations." in Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery, 2010, pp. 2461–2464.

[19] J. Wang, N.-Y. Zhang, J. Yin, J, "Collaborative filtering recommendation based on fuzzy clustering of user preferences," in Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery, 2010, pp. 1946–1950.

[20] D. –R. Liu, Y. –Y. Shih, "Hybrid approaches to product recommendation base on customer lifetime value and purchase preferences," Journal of Systems and Software, vol. 77, no. 22, pp. 181–191, 2005.

[21] O. Georgiou and N. Tsapatsoulis, "Improving the scalability of recommender systems by clustering using genetic algorithms," in Proceedings of the 20th Int. conf. Artificial Neural Networks, 2010, pp. 442–449.

[22] K. Honda, N. Sugiura, H. Ichihashi and S. Araki, "Collaborative filtering using principal component analysis and fuzzy clustering," in Proceedings of the 20th International Conference on Artificial Neural Networks: Part I, 2001, pp. 442–449.

[23] A. Selamat and S. Omatu, "Web page feature selection and classification using neural networks," Information Science, vol.158, pp. 69–88, 2004.

[24] J. Han and M. Kamber, Data Mining: Concepts and Techniques. San Francisco CA: Morgan Kaufmann Publishers, 2001.

[25] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principle. Massachusetts: Addison Wesley, 1974.

[26] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. New York: Addison-Wesley, 1989.

[27] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proceedings of the 1994 Computer Supported Cooperative Work Conference, 1994, pp. 175–186.