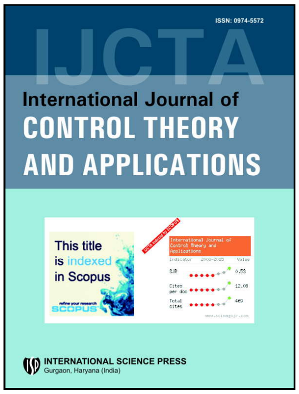


International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 18 • 2017

Hybrid Movie Recommendation System based on PSO based Clustering

Rabi Narayan Behera[1], Progga Laboni Saha[2], Anindita Chakraborty[2] and Sujata Dash[1]

[1] Department of Computer Science & Application North Orissa University, Baripada, India,
Emails: rabi.behera@iemcal.com, sujata238dash@gmail.com
[2] Department of Information Technology, Institute of Engineering & Management, Kolkata, India,
Emails: progga2009@gmail.com, me_anindita1990@rediffmail.com

Abstract: Due to rapid digitization user shows interest for finding suggestions or recommendations regarding a particular topic before taking decision. Recommender system is applied to some popular areas like music, videos, movies, books, news, articles etc. Recommendation System predict the "rating" or "preference" that a user would give to an item based on information filtering from the web. Nowadays, a movie recommender system is an important area which suggests movie based on user profile. In this paper we introduce movie recommendation system based on PSO based clustering which developed a model on ensembeled supervised machine learning for prediction and PSO based cluster analysis based on a hybrid recommendation approach which includes both collaborative and content based filtering to design a profile matching algorithm.

Keywords: Recommender system, K-means, SVM, Collaborative filtering, Content based filtering, Hybrid filtering, Particle Swarm Optimization.

1. INTRODUCTION

With the explosion of web 2.0, people are more fascinated toward internet and faces problem for searching of personalized items from a huge number of choices. Everyone is rushing to achieve ultimate goals. This thirst results into the development of almost every sector. Many companies have recommendation system to help their users. Recommendation system also helped providers in several ways like additional and probably unique personalized service for the customer, increase of trust and customer loyalty, increase of sales, clicking of trough rates, creating opportunities for promotion and persuasion, obtaining more knowledge about customers, etc. A "recommender system" is a fully automated functional software to make recommendations of items by applying at least one implementation [1]. In addition, feature of recommender system are consists of several other components, like user interface, recommendation candidates corpus, and an automated operator that runs the system. Some recommender systems use multiple recommendation approaches like CiteULike, a service for discovering and managing scholarly references [2][3].

Recommendation system is an ongoing research area for last few decades but it needs more attention due to the abundance of practical applications point of view. Some popular website like Amazon.com for book, MovieLens.org for movies and CDNow.com for CD uses recommendation system for personalized recommendation. Recommendation System is seen as a function in which input is given in the form of user model (like ratings, preferences, demographics, situational context) or items (with or without characteristics description of an item) to find the relevant score or is used for ranking. Finally Items are recommended assumed to be relevant but the relevance might be influenced by context and diversity on list of characteristics. Recommendation system as a silent and integral part of some e-commerce websites like Amazon.com, Netflix etc helps in their economic growth [4]. A glimpse of the profit of some websites is shown in table below:

Netflix	2/3rd of the movies watched are recommended
Google news	recommendations generate 38% more click-troughs
Amazon	recommendations enhances 35% sale
Choicestream	28% of the people would buy more music if they found what they liked

We can classify the recommender systems in three categories:

1. Collaborative filtering approach
2. Content-based filtering approach
3. Hybrid filtering approach

1.1. Collaborative filtering

Collaborative filtering (CF) is a ratings or ranking based recommendation algorithm with a fundamental assumption that selection and aggregation of other user's opinion in order to predict active user's preference more accurately and reasonably. It based on the assumption that, if a user agrees about the quality or relevance of some items, then they will likely to agree about other items liked by the users of the same group. Some social networking site such as LinkedIn and Facebook etc use collaborative filtering to recommend new friends, groups and other social connections. There are some advantages of collaborative system like

Collaborative do not require any content information.

This system is mainly used to make an assessment of quality, style or view point from other people's experience. It can suggest serendipitous items.

Algorithm:

```
Given a user x, recommend a movie that x might like
For all movies m set score(m)=0 for each other user y
If y's preferences match x's preferences then increment score(m) for all m that y likes
Find the movie with the highest score and return it.
```

The major challenges of collaborative filtering are:

Cold start problem- it happens when insufficient opinions are available for a newly joined user with the existing one due to lack of interactive communication, and hence it is not possible to measure the similarity between them. As a result, the recommender systems are unable to predict any inferences for users. [4]

Sparsity-In the present scenario, some commercial recommender systems such as Amazon.com,CDnow.com etc. are used to evaluate large set of items. In these systems, even active users may have purchased well under 1%

of the items (1% of 2 million books is 20,000 books). A recommender system based on nearest neighbor algorithms may not predict proper items for a specific user which leads to poor accuracy.

Scalability- Computational requirement of Nearest Neighbor algorithms proportionally increases with the increasing in number of users and items. Some Web based recommender system suffers from higher degree of scalability.

1.2. Content based filtering

Content based filtering makes prediction based on the profile of the user's preference and the description of item's characteristics. In CBF to describe items we use keywords apart from user's profile to indicate user's preferred liked or dislikes. In other words CBF algorithms recommend those items or similar to those items that were liked in the past. It examines previously rated items and recommends best matching item .Some advantages of Content based filtering are as follows

Content-based method gives better transparency because content-based method recommends the items based on their features.

Content based filtering is user independent. If any user wants to write a content he may or may not be dependent on other user's content.

Algorithm:

```
Given a movie m1, find the most similar movie-
Set best score to some minimum value (0)
For each other movie m2 set match score=0
For each feature compare m1 and m2 on that feature
If they match, increment match score
If match score > best score then best match=m2 and best score=match score and return best match
Content based filtering also has some problems-
```

Limited Content Analysis- Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend. Domain knowledge is often needed, e.g., for movie recommendations the system needs to know the actors and directors, and sometimes, domain ontology are also needed. Content based recommendation system can't provide suitable suggestions if the analyzed content does not contain enough information to discriminate items the user likes from items the user does not like.

Over-Specialization –A content based filtering is also suffer from over specialization and can't select items if the previous user behavior does not provide evidence for this.

New User - Reliable recommendations can be provided for new user if enough ratings are available so that a content-based recommender system can really understand user preferences and provide accurate recommendations.

1.3. Hybrid filtering

In Hybrid filtering more than one filtering approaches are combined .Hybrid filtering used to overcome some problems associated with collaborative and content-based filtering such as sparsity problem, cold start problem, overspecialization problem etc. We can implement Hybrid filtering approach to get more accurate and efficient recommendation system .There is various ways to implement Hybrid approach:

1. Collaborative and content-based method can be generated individually, and then we can aggregate their predictions.
2. Some content-based characteristics can be integrated into a collaborative approach.
3. Some collaborative characteristics can be comprised into a content-based approach.
4. We can integrate both content-based and collaborative approach by constructing a general consolidative model.

When we integrate CBF (content-based) characteristics into the CF (collaborative) approach then the cold start problem is overcome in collaborative filtering and the overspecialization problem of content-based filtering is also overcome .To improve the effectiveness of recommendation process ,we can construct a unified utility system by combining some features of CBF and CF.

When no training information is available (cold start) in case of collaborative filtering then content-based filtering systems are used to overcome this problem, but the result is less accurate compare to collaborative filtering. In case of hybrid recommendation schemes these different kinds of information are aggregated to get efficient recommendation result.

2. RELATED WORK

In the mid of 1990s, recommender systems have become a popular research area for many researcher [5]. These recommendation systems use a variety of methods such as content based approach, collaborative approach, knowledge based approach, utility based approach, hybrid approach, etc. A personalized recommendation system to suggest new products in supermarkets to shoppers based on their previous purchase behavior was described by Lawrence et al. 2001. Linden, Smith and York designs an algorithm called item-to-item collaborative filtering for recommendation in real time, is applicable to massive datasets and produces fine quality recommendations (Linden, Smith, & York, 2003). In 2006 Hybrid Approach based on CF and Neural Network using movielens dataset was proposed [6]. In 2007 Web based movie recommender system was presented, which of the three techniques Demographic filtering, Content-based filtering, and Collaborative filtering has been used[7]. In 2007 Weng, Lin and Chen performed an evaluation study which says using multidimensional analysis and additional customer's profile increases the recommendation quality. Weng used MD recommendation model (multidimensional recommendation model) for this purpose. Multidimensional recommendation model was proposed by Tuzhilin and Adomavicius (2001).

3. THE PROPOSED HYBRID APPROACH

In our previous work [9] we have already discussed about particle swarm optimization based hybrid recommendation system. Now, we are trying to modify previous approach to get better analysis. We have already seen that collaborative and content based filtering has some limitations for movie recommendation system. This paper proposed a new hybrid movie recommender system, by which higher accuracy of the prediction will be achieved. In fact we combined multiple recommendations techniques to improve a movie recommender system. Integrated hybrid scheme can be modeled by combining both Collaborative and content-based filtering [8]. Different types of data that will be applied to recommender systems are content based information and collaborative information. We used Movie Lens dataset which consists of many users and many features and it is used for both training and testing purpose. When a new user registers, his age, gender, occupation must be collected and depending upon these features, system can predict his movie preference. Users and the movie's information are stored in Movie Lens dataset where user information includes age, gender, occupation of the user and movie information includes movie name, release date, and genres. Both PSO and GA use 21 features from dataset: age, gender, occupation and 18 genres. The step by step designing of the system is illustrated below-

3.1. Clustering of sample

We use k-means for clustering of sample to group unsupervised data. K-means is used to get the best profile matching between two users.

1) At first we select no of cluster which is known to be k .for an example if we select k=10 that mean 10 groups will be created.

2) Select k no of cluster centers such that they are far away from each other.

3) Now consider each data point and assigned it to the closest cluster.

4) Recalculate cluster centers which can be found by mean of data point belonging to the same cluster.

5) Repeat step 3 and 4 till shifting of cluster centers are observed.

So, in our approach for movie recommendation we choose user's age to predict the most similar profile. These age can be taken as random manner and then we apply k-mean algorithm which creates group belonging to the same age range.

K -means algorithm used for reducing an object function, which is a square error function. The objective function is:

$$j = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Where j = Objective function

k = no of clusters

n = no of cases

x_i = case i

c_j = centroid for cluster j

$\left\| x_i^{(j)} - c_j \right\|^2 = \textit{distance function}$

3.2. SVM Classification

SVM is the efficient machine learning classification tool that describes correlation between personal particulars of a user and his personal preferences for movies. For each user input, SVM is trained based on a predefined set of training samples. Machine learning is a process of training and predicting the dataset. Those data act as the prior knowledge of the specific domain of interest. Usually for supervised machine learning problems, the training data are composed of both input and output parts. In our system, we have two sets of data, the training data set and the testing data set comprises of different training and testing samples. Each of which contains an input vector, while the output vector is to be predicted by the machine learning. We have used k-means for clustering the sample which is called training data. When a new user wants to register then the testing data matched with the trained data like user's age occupation, gender etc are matched with the existing clusters then it will decide which cluster it can be matched.

3.3. Particle Swarm Optimization

Nature-Inspired Optimization Algorithms offers a systematic overview to all nature based algorithms for optimization. There are many nature inspired algorithms are available like particle swarm optimization, ant and bee algorithms, simulated annealing, cuckoo search, firefly algorithm, bat algorithm, harmony search etc, Nature based algorithm used for weighting and ranking the feature construction and dimensionality reduction.

By using k-mean we have found threshold value for each group. The PSO algorithm has been used to fine-tune the profile matching process to provide more accurate results. The profile generation process followed by the profile selection and matching is further polished by the use of PSO. User's preference or taste can be changed according to their age variation. Suppose user A likes Horror movie at the age of 20 but when he turns to 25 his taste may be changed. At this point we cannot say that he will belong to the same group or cluster before 5 years back he was. So we need optimization techniques to update his cluster. PSO is the nature based algorithm which was inspired by social behavior of birds and fishes. It was proposed by James Kennedy & Russell Eberthart (1995)."Swarm" means collection of disorganized moving individual or particles and each particle moving in a random direction to find the optimal solution.

In search space each particle adjusts its "flying" according to its own flying experience as well as the flying experienced of other particles.

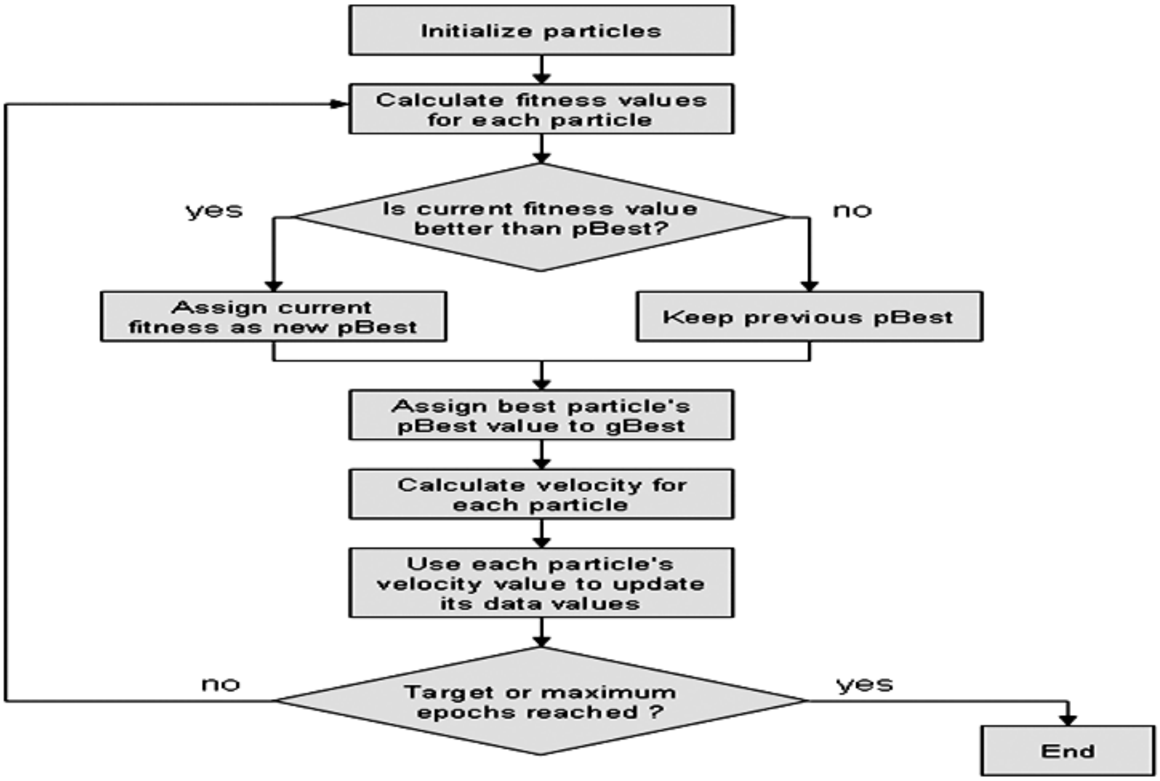
Particles are initialized by random position: $x_i = \left(x_{i,1}, x_{i,2} ......, x_{i,n}\right) \in R^n$ and

Velocity: $v_i = \left(v_{i,1}, v_{i,2} ......, v_{i,n}\right) \in R^n$ where where xi is the current position of particle i. v_i is the velocity of particle i

Each particle keeps track of its best position in hyperspace.

"pbest" is used for an individual particle, "gbest" for the global best population and "lbest" for local best neighbor.

Each step, each particle stochastically accelerates towards its pbest, gbest or lbest.


Initialize particles
Calculate fitness values for each particle
Is current fitness value better than pBest?
yes
no
Assign current fitness as new pBest
Keep previous pBest
Assign best particle's pBest value to gBest
Calculate velocity for each particle
Use each particle's velocity value to update its data values
Target or maximum epochs reached ?
no
yes
End

The main approach of PSO is to accelerate its particle towards its pbest and gbest.

PSO velocity updates equation-

$$v_{id} = w * v_{id} + c1 * rand_1 (\ ) * \left(p_{id} - x_{id} \right) + c_2 * rand_2 (\ ) \left(p_{gd} - x_{id} \right) x_{id} = x_{id} + v_{id}$$

Where w denotes the inertia weights, d is the dimension, c1, c2 are positive constant, rand1, rand2 are random functions in the scope of(0,1). For lbest change p_{gd} to p_{id}.

Best fitness value can be computed as-

$$Pi\left(t+1\right) = Pi\left(t\right) f\left(Xi\left(t+1\right) \le f\left(Xi\left(t\right)\right) Xi\left(t+1\right) f\left(Xi\left(t+1\right)\right) > f\left(Xi\left(t\right)\right)\right)$$

Here, f represents fitness function, Pi(t) finest fitness function, t is the generation step.

For movie recommendation system PSO algorithm is-

```
1.  for each particle i = 1, ..., S do
2.  Initialize the particle's position with uniformly distributed random vector: xi ~ U(blo, bup)
3.  Initialize the particle's best known position to its initial position: pi ← xi
4.  if f(pi) < f(g) then
5.  update the swarm's best known position: g ← pi
6.  Initialize the particle's velocity: vi ~ U(-|bup-blo|, |bup-blo|)
7.  while a termination criterion is not met do:
8.  for each particle i = 1, ..., S do
9.  for each dimension d = 1, ..., n do
10. Pick random numbers: rp, rg ~ U(0,1)
11. Update the particle's velocity: vi, d ← ω vi, d + φp rp (pi, d-xi, d) + φg rg (gd-xi, d)
12. Update the particle's position: xi ← xi + vi
13. if f(xi) < f(pi) then
14. Update the particle's best known position: pi ← xi
15. if f(pi) < f(g) then
16. Update the swarm's best known position: g ← pi
```

3.4. Fitness Function

The position values are used to calculate a set of feature weights. The given factor, the weight reduction size, is used to reduce the importance of 18 genre frequencies which can be categorized under the single large feature called the Genre. This reduction is done to provide an equal amount of importance to the other unrelated features (rating, age, gender, occupation). Next the total value of the position is calculated by taking the sum of the position values on all the 22 axes. The weighting value for each feature is found by dividing the real value by the total value. The sum of all the weights adds up to unity. Every particle's current position (set of weights) in the swarm must be given by the profile matching process in the recommender system. So the recommender system is re-run on the dataset for each new position and calculates the fitness score.

4. EXPERIMENT AND RESULT

We are going to describe the experimental setup and related work of this movie recommendation system with PSO algorithm.

Here, we have taken Movie Lens dataset to consider both training and testing purpose which is easily available dataset and it assigned a discrete scale of 1-5 for rating a movie. we get user personal details like age, gender and occupation from the user data set, the movie title and its generic into 19 types are described in the item dataset movie details and the rating given by a user on a specific movie is known from the rating dataset .Then we have to consider the user details as well as movie details to generate a new dataset.

At first we have to cluster the users in different groups according to their age. Here, we consider the users from 10 to 60 years and they are clustered into 10 groups. Then to find a better correlation between 2 users into same age group, the user who rated less than 8 will be discarded and further elimination will be take place based on the number of common movies rated by 2 users which is less than 4.

For a particular movie user can rate on a scale of 1 to 5. First time we use K-mean for clustering the user according to their age. If two users belong to same age group that time we again use K-mean algorithm to find the rating differences between 2 users rated some common movies. According to K-mean we get 10 different clusters where user can be fitted.

Age	*Gender*	*Occupation*	*18*	*Genre frequencies*	*Class Label*
25	1	1	000010000010000000	C3	

The K-NN function measures the similarity index between any two existing users for all the users. All the 22 features values of each generated profiles are taken into consideration. To consider or ignore the features, every feature is assigned with a binary weight. Any 2 users having same gender will be assigned with 1, otherwise 0. The value "1" indicates consideration while "0" to be discarded. The K-NN distances between two users that have the nearest values are fitted into the same cluster. Based on the available user, movie and review data, users were grouped into 10 clusters. Now, a new labeled dataset was prepared with some fields like user profile, movie name, and its class names as cluster names. The dataset was split into training and test data in the ratio of 9:1 respectively. The training data was used to learn the model and the test data to evaluate the accuracy of the predictions. Here we use SVM classifier is used to train the model. Then PSO is used to tune or update the user profile by merging different particles to form strong particles where particle represents cluster centroids which provide a hierarchical solution for clustering problem.

5. CONCLUSION AND FUTURE SCOPE

In this paper, proposed hybrid recommendation system based on SVM classifier on Movie Lens dataset with K-means clustering is presented and observed that our proposed methods perform better than collaborative and content based approach. It is important to adopt other types of datasets to verify our methodology, because our experiments used only one dataset. Another important point is that User's preference or taste can be changed according to their age variation. Suppose user A likes Horror movie at the age of 20 but when he turns to 25 his taste may be changed. At this point we cannot predict that he will belong to the same group or cluster before 5 years back he was. it is to be seen how the notion of contexts can be incorporated in the proposed approach for further improvement.

REFERENCES

[1] Diao, Qiming, et al. "Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars)."Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining ACM, 2014.

[2] T. Bogers and A. van den Bosch, "Recommending scientific articles using citeulike," in Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 287–290

[3] Beel, Joeran, Stefan Langer, and Andreas Lommatzsch. "Exploring the Butterfly Effect and (Non-) Reproducibility in Recommender Systems Research."

[4] Vala Ali Rohani,Zarinah Mohd Kasirun, Sameer Kumar,and Shahaboddin Shamshirband." An Effective Recommender Algorithm for Cold-Start problem in Academic Social Networks

[5] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-ofthe-Art and Possible Extensions," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO.6, pp734- 749, 2005

[6] Vassiliou, Charalampos, et al. "A recommender system framework combining neural networks collaborative filtering." Proceedings of the 5th WSEAS international conference on Instrumentation, measurement, circuits and systems. World Scientific and Engineering Academy and Society (WSEAS), 2006

[7] Nguyen, Ngoc Thanh, et al. "Hybrid filtering methods applied in web-based movie recommendation system." International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer Berlin Heidelberg, 2007.

[8] Analysis and Design of Hybrid Online Movie Recommender System.Harpreet Kaur Virk Department Of Computer Science Chandigarh University Gharuan,India.Er.Maninder Singh Department Of Computer Science Chandigarh University Gharuan, India.Er. Amritpal Singh Thapar University Patiala, India

[9] Rabi Narayan Behra, Sujata Dash. "A Particle Swarm Optimization based Hybrid Recommendation System." International Journal of Knowledge Discovery in Bioinformatics (IJKDB) 6.2 (2016): 1-10.

[10] Kumar, Manoj, et al. "A movie recommender system: Movrec." International Journal of Computer Applications 124.3 (2015).