# A Movie Recommender System: MOVREC

**Article** *in* International Journal of Computer Applications · August 2015

DOI: 10.5120/ijca2015904111



**4 authors**, including:

Manoj Kumar
Motilal Nehru National Institute of Technology
**5** PUBLICATIONS **66** CITATIONS

SEE PROFILE

Dharmendra Kumar Yadav
Motilal Nehru National Institute of Technology
**51** PUBLICATIONS **294** CITATIONS

SEE PROFILE

# A Movie Recommender System: MOVREC

Manoj Kumar
Assistant Professor
Department of Information and Technology
BBDNITM Lucknow

D.K. Yadav
Associate Professor
Department of Computer Science
MNNIT Allahabad
Allahabad

Ankur Singh
M.Tech (P)
Software Engineering
BBDU Lucknow
India

Vijay Kr. Gupta
Assistant Professor
Department of Computer Science
BBDNITM Lucknow

## ABSTRACT
Now a day's recommendation system has changed the style of searching the things of our interest. This is information filtering approach that is used to predict the preference of that user.

The most popular areas where recommender system is applied are books, news, articles, music, videos, movies etc. In this paper we have proposed a movie recommendation system named MOVREC. It is based on collaborative filtering approach that makes use of the information provided by users, analyzes them and then recommends the movies that is best suited to the user at that time. The recommended movie list is sorted according to the ratings given to these movies by previous users and it uses K-means algorithm for this purpose. MOVREC also help users to find the movies of their choices based on the movie experience of other users in efficient and effective manner without wasting much time in useless browsing. This system has been developed in PHP using Dreamweaver 6.0 and Apache Server 2.0. The presented recommender system generates recommendations using various types of knowledge and data about users, the available items, and previous transactions stored in customized databases. The user can then browse the recommendations easily and find a movie of their choice.

## Keywords
K-means, recommendation system, recommender system, data mining, clustering, movies, Collaborative filtering, Content-based filtering

## 1. INTRODUCTION
In today's world where internet has become an important part of human life, users often face the problem of too much choice. Right from looking for a motel to looking for good investment options, there is too much information available. To help the users cope with this information explosion, companies have deployed recommendation systems to guide their users. The research in the area of recommendation systems has been going on for several decades now, but the interest still remains high because of the abundance of practical applications and the problem rich domain. A number of such online recommendation systems implemented and used are the recommendation system for books at Amazon.com , for movies at MovieLens.org, CDs at CDNow.com (from Amazon.com), etc.

Recommender Systems have added to the economy of the some of the e-commerce websites (like Amazon.com) and Netflix which have made these systems a salient parts of their websites. 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 | 35% sales from recommendations |
| Choicestream | 28% of the people would buy more music if they found what they liked |

**Table1. Companies benefit through recommendation system**

Recommender Systems generate recommendations; the user may accept them according to their choice and may also provide, immediately or at a next stage, an implicit or explicit feedback. The actions of the users and their feedbacks can be stored in the recommender database and may be used for generating new recommendations in the next user-system interactions. The economic potential of theses recommender systems have led some of the biggest e-commerce websites (like Amazon.com, snapdeal.com) and the online movie rental company Netflix to make these systems a salient part of their websites. High quality personalized recommendations add another dimension to user experience. The web personalized recommendation systems are recently applied to provide different types of customized information to their respective users. These systems can be applied in various types of applications and are very common now a day.

We can classify the recommender systems in two broad categories:

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

## 1.1 Collaborative filtering
Collaborative filtering system recommends items based on similarity measures between users and/or items. The system recommends those items that are preferred by similar kind of users. Collaborative filtering has many advantages

1. It is content-independent i.e. it relies on connections only
2. Since in CF people makes explicit ratings so real quality assessment of items are done.
3. It provides serendipitous recommendations because recommendations are base on user's similarity rather than item's similarity.

## 1.2 Content-based filtering

Content-based filtering is based on the profile of the user's preference and the item's description. 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.

There are various approaches proposed in various research papers listed below. These approaches are often combined in Hybrid Recommender Systems. An earlier study by Eyjolfsdottir *et. al* for the recommendation of movies through MOVIEGEN had certain drawbacks such as , it asks a series of questions to users which was time taking . On the other hand it was not user friendly for the fact that it proved to be stressful to a certain extent. Keeping in mind these shortcomings, we have developed MovieREC, a movie recommendation system that recommends movies to users based on the information provided by the users themselves. In the present study, a user is given the option to select his choices from a set of attributes which include actor, director, genre, year and rating etc. We predict the users choices based on the choices of the previous visited history of users. The system has been developed in PHP and currently uses a simple console based interface.

# 2. RELATED WORK

Many recommendation systems have been developed over the past decades. These systems use different approaches like collaborative approach, content based approach, a utility base approach, hybrid approach etc.

Looking at the purchase behavior and history of the shoppers, Lawrence et al. 2001 presented a recommender system which suggests the new product in the market. To refine the recommendation collaborative and content based filtering approach were used. To find the potential customers most of the recommendation systems today use ratings given by previous users. These ratings are further used to predict and recommend the item of one's choice.

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. RESEARCH METHODOLOGY

## 3.1 The Basic K-means Algorithm

The original K-means algorithm was proposed by MacQueen [20] .The ISODATA algorithm by Ball and Hall[22] was an early but sophisticated version of k-means. Clustering divides the objects into meaningful groups. Clustering is unsupervised learning. Document clustering is automatic document organization.

In K-means clustering technique we choose K initial centroids, where K is the desired number of clusters. Each point is then assigned to the cluster with nearest mean i.e. the centroid of the cluster. Then we update the centroid of each cluster based on the points that are assigned to the cluster. We repeat the process until there is no change in the cluster center (centroid). Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{j=1}^{n} \left\| x_i^{(j)} - C_j \right\|^2 \quad (1)$$

Where, k is the number of clusters, n is the number of cases $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre $c_j$, is an indicator of the distance of the n data points from their respective cluster centers.

The algorithm is composed of the following steps:

1. Select K points as initial centroids.
2. Repeat
3. From k clusters by assigning each point to its closest centroid.
4. Re-compute the centroid of each cluster.
5. Until Centroid do not change.

**Figure 1. K-means Algorithm**

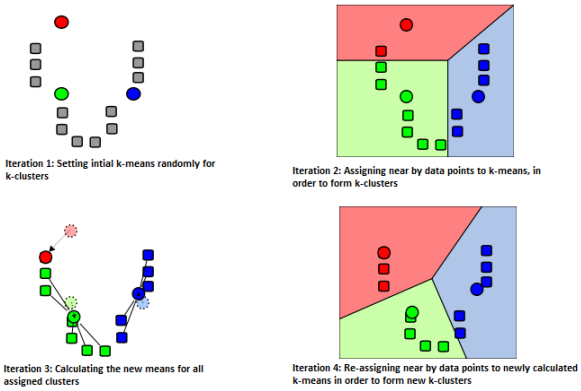


**Figure 2: The 5 steps of the K-Means algorithm [3]**

## 3.2 Data Description

In proposed model we use a pre filter before applying K-means algorithm. The attributes used to calculate distance of each point from centroid are

1. Genre
2. Actor
3. Director
4. Year
5. Rating

Different attributes have different weights. In our research we have found that the most appropriate recommendations that can be generated should be based on the ratings given to the movies by previous users, therefore we have given more importance to the rating attribute than other attributes. These ratings have been taken from www.imdb.com because perhaps it has the largest collection of movies along with the rating given to these movies by a large number of different users from different parts of the world. Another important parameter in our proposed model is total number of votes received by a particular movie. We have divided number of votes in to three categories that is less than or equal to 1000, more than 1000 but less than or equal to 10,000 and greater than 10,000.

In our research we have found that as the number of vote's increases the weight of rating should also increase respectively. Therefore we have used ratios of 1:1, 1:2, and 1:3 depending on total number of votes received by a movie. we have also found that the movies which have rating less than 5 are the ones which are least suitable for recommendation, and are least desirable by users. Users generally want to see a good movie and higher rating ensures that our predicted movie set are of those movies which are liked by a large number of users. Weights assigned to other attributes are generally based on the average of total movies associated with that particular attribute to the total number of movies in our data set.

## 3.3 Simulation of MOVREC

When any user enters our system MOVREC he has a couple of options. He /she can search a particular movie or see upcoming movies list or can go to our recommendation page. On recommendation page he is given the choice to select/input values for different attributes. On the basis of these input values, we search our search our database and prepare an array of suitable movies. Movies included in the array are those whose even one attribute value matches with the input value of the user. We then calculate the number of movies in our array with the help of a counter. If the counter value is less than or equal to twenty we display the movie list sorted according to ratings associated with the movies. If number of movies is greater than twenty then we apply a pre filter and select top twenty movies according to rating. If two movies have same rating then priority is given to the movie having a large number of votes. After filtering the movie list we match the attributes value to their respective weights and compute the total weight of each movie. Once we have calculated the total weight of each movie we apply K-means clustering algorithm on these group of movies. In our research we have also found that generally a user prefer a list with five movies so we assume K equal to be 4 so that an average every K has five movies, where K is the number of cluster to be formed.

For each cluster k1, k2 , k3, k4 we assume initial centroid c1, c2, c3, c4 which corresponds to the first, sixth, eleventh, and sixteenth movie in the movie array. After defining the initial centroid we compute the distance of all the other data points from each centroid and assign the remaining data points (movies) to closest centroid and form clusters. The distance measure we have used to calculate the distance between data points and centroid is the Euclidean Distance.

After forming initial clusters we take one cluster at a time. We again calculate centroids but this time each centroid corresponds to mean of the points in that cluster. After recalculating centroids we compute the distance of all data points with respect to these newly formed centroids and reassign them to form clusters. We repeat this process till there is no change in centroids. This ensures that the clusters finally formed are optimized and no further grouping is possible. Once final cluster are formed we compute the average rating of all points belonging to that cluster i.e. cluster rating, then according to the input user query we display the cluster having highest cluster rating.

### *Weightage and matching of attributes*

1. **Actor ($W_a$)**

$$W_a = \frac{\text{No. of movies of Actor(a) in data set}}{\text{Total no. of movies in data set}}$$

2. **Director ($W_d$)**

$$W_d = \frac{\text{No. of movies of Director(d) in data set}}{\text{Total no. of movies in data set}}$$

3. **Rating**

| Rating | Weight | | |
|---|---|---|---|
| | **If number of votes <=1000** | **If number of 1000< votes <=10000** | **If number of votes >10000** |
| 10 | 10 | 20 | 30 |
| 9 | 9 | 18 | 27 |
| 8 | 8 | 16 | 24 |
| 7 | 7 | 14 | 21 |
| 6 | 6 | 12 | 18 |
| 5 | 5 | 10 | 15 |
| 1-4.9 | 1 | 2 | 3 |

4. **Genre($W_g$)**

$$W_g = \frac{\text{No. of movies of Genre(g) in data set}}{\text{Total no. of movies in data set}}$$

5. **Year($W_y$)**

$$W_y = \frac{\text{No. of movies in Year (Y) in data set}}{\text{Total no. of movies in data set}}$$

Total weight of a particular movie m is given by

$$W_m = W_r + W_a + W_d + W_g + W_y$$

## 3.4 Proposed Algorithm

**Input:** a number of movies: m

**Output:** a number of clusters: K

**Step 1** Select n movies from m movies n<m

**Step 2** If n>20 then select top 20 movies from n movies based on ratings.
Else display the output movies sorted by rating.

**Step 3** If rating of movies x, y are equal i.e. If $R_x = R_y$
Then select those movies which have greater number of user votes.

**Step 4** Assume K=4.

**Step 5** REPEAT (6, 7)

**Step 6** Chose initial centroid $C_1, C_2, C_3, C_4$.

**Step 7** Calculate Euclidean distance of all data points w.r.t. $C_1, C_2, C_3, C_4$ and re-compute the centroid of each cluster.

**Step 8** UNTILL centroid does not change.

Where,

**m**: Total number of movies in database
**n**: Number of movies after user query
**x, y**: Two random movies
$\mathbf{R_x}$, $\mathbf{R_y}$: Rating of movies x, y
**K**: Number of cluster
$\mathbf{C_1}$, $\mathbf{C_2}$, $\mathbf{C_3}$, $\mathbf{C_4}$: Initial Centroid.

## 3.5 Challenges Faced

In developing any system the biggest challenge is to satisfy the end users for which the system is being developed. We also faced certain challenges while developing our system. Some of them are:

- To have a system that is user friendly and easy to understand and use.
- To create a data set that has all relevant information about a particular movie.
- The biggest challenge was to have the most appropriate movie recommended list.
- To make our system diversifiable so that it can satisfy users of different geographical locations.
- To give weights to different attributes.

## 3.6 Overcome the problems

- The proposed system has been tested over a small group of people, and we have received a positive response from them. We have kept our system simple and interactive for this we have choose php and java script.
- For collecting information we have intensively search free online movie data bases and extract the information which was useful for our proposed system.
- To accurately recommend movie to user we have applied K-means clustering algorithm along with a pre filter.
- We have included movies in our database irrespective of their language or location so that users from all across the globe can use our system.
- For assigning weights to attributes and for giving priority to them we have conducted a survey on a group of people and on the basis of the result obtained we have prioritize our attributes.

*An excerpt of the survey is given below*

| Name | Actor/ Cast | Genre | Rating | Director | Year |
|---|---|---|---|---|---|
| Mrs Malti Singh | 2 | 5 | 1 | 4 | 3 |
| K.K Singh | 5 | 3 | 1 | 2 | 4 |
| Kritika Baranwal | 1 | 3 | 2 | 5 | 4 |
| Ranjana Rai | 3 | 2 | 1 | 4 | 5 |
| Dr. Siddharth Singh | 2 | 3 | 1 | 4 | 5 |
| Alok Kumar | 1 | 4 | 2 | 5 | 3 |
| Shashank Pandey | 3 | 2 | 1 | 5 | 4 |
| Er. Pankaj Agarwal | 3 | 2 | 1 | 4 | 5 |
| Mr. Girjesh Mishra | 3 | 1 | 2 | 4 | 5 |

**Table 2 . Excerpt of the Survey Conducted**

In our survey we have found that almost every user has given maximum priority to the rating concerned with the movie so we have given the maximum priority to our rating attribute. Also we have deduced from our survey that movie rating has more weight if that movie has received a large number of votes. So we have divided rating and votes in to three categories i.e. minimum, medium and maximum votes.

# 4. CONCLUSION

In this paper we have introduced MovieREC, a recommender system for movie recommendation. It allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes and using K-means algorithm. By the nature of our system, it is not an easy task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinions. Based on informal evaluations that we carried out over a small set of users we got a positive response from them. We would like to have a larger data set that will enable more meaningful results using our system. Additionally we would like to incorporate different machine learning and clustering algorithms and study the comparative results. Eventually we would like to implement a web based user interface that has a user database, and has the learning model tailored to each user.

# 5. REFERENCES

[1] Han J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann (Elsevier), 2006.

[2] Ricci and F. Del Missier, "Supporting Travel Decision making Through Personalized Recommendation," Design Personalized User Experience for e-commerce, pp. 221-251, 2004.

[3] Steinbach M., P Tan, Kumar V., "Introduction to Data Mining." Pearson, 2007.

[4] Jha N K, Kumar M, Kumar A, Gupta V K "Customer classification in retail marketing by data mining" International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014 ISSN 2229-5518

[5] Giles C.L., Bollacker K.D., and Lawrence S., "CiteSeer: An automatic citation indexing system," in Proceedings of the third ACM conference on Digital libraries, 1998, pp. 89–98.

[6] Beel J., Langer S., Genzmehr M., and Nürnberger A., "Introducing Docear's Research Paper Recommender System," in Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'13), 2013, pp. 459–460.

[7] Bethard S and Jurafsky D, "Who should I cite: learning literature search models from citation behavior," in Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 609–618.

[8] Bollacker K. D., Lawrence S., and Giles C. L., "CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications," in Proceedings of the 2nd international conference on Autonomous agents, 1998, pp. 116–123.

[9] Erosheva E., Fienberg S., and Lafferty J., "Mixed-membership models of scientific publications," in Proceedings of the National Academy of Sciences of the United States of America, 2004, vol. 101, no. Suppl 1, pp. 5220–5227.

[10] Ferrara F., Pudota N., and Tasso C., "A Keyphrase-Based Paper Recommender System," in Proceedings of the IRCDL'11, 2011, pp. 14–25.

[11] Jiang Y., Jia A., Feng Y., and Zhao D., "Recommending academic papers via users' reading purposes," in Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 241–244.

[12] McNee S. M., Kapoor N., and Konstan J. A., "Don't look stupid: avoiding pitfalls when recommending research papers," in Proceedings of the 20th anniversary conference on Computer supported cooperative work, 2006, pp. 171–180.

[13] Middleton S. E., De Roure D. C., and Shadbolt N. R., "Capturing knowledge of user preferences: ontologies in recommender systems," in Proceedings of the 1st international conference on Knowledge capture, 2001, pp. 100–107.

[14] Zarrinkalam F. and Kahani M., "SemCiR - A citation recommendation system based on a novel semantic distance measure," Program: electronic library and information systems, vol. 47, no. 1, pp. 92–112, 2013.

[15] Schafer J. B., Frankowski D., Herlocker J., and Sen S., "Collaborative filtering recommender systems," Lecture Notes In Computer Science, vol. 4321, p. 291, 2007.

[16] Seroussi Y., "Utilising user texts to improve recommendations," User Modeling, Adaptation, and Personalization, pp. 403–406, 2010.

[17] Buttler D., "A short survey of document structure similarity algorithms," in Proceedings of the 5th International Conference on Internet Computing, 2004.

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

[19] Beel J., Langer S., and Genzmehr M., "Mind-Map based User Modelling and Research Paper Recommendations," in work in progress, 2014.

[20] MacQueen J.. Some methods for classification and analysis of multivariate observations. In Proc. Of the 5th Berkeley Symp. On Mathematical Statistics and Probability, pages 281-297. University of California Press, 1967.

[21] Ball G. and Hall D.. A Clustering Technique for Summarizing Multivariate Data. Behavior Science, 12:153-155, March 1967. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.