Impact of Similarity Measures in K-means Clustering Method used in Movie Recommender Systems

Mayur Rahul¹, Parashuram Pal², Vikash Yadav³, Devendra Kumar Dellwar⁴ and Swarnima Singh⁵

¹ Department of Computer Application, UIET CSJM Kanpur, Uttar Pradesh, India
² ³ ⁵ ABES Engineering College, Ghaziabad, Uttar Pradesh, India
⁴ SR Group of Institution, Jhansi, Uttar Pradesh, India
¹mayurrahul209@gmail.com, ²parashuram@abes.ac.in, ³vikash.yadav@abes.ac.in,
⁴devendra.kumar6@gmail.com, ⁵sswarnima.singh@gmail.com

Abstract: One of the most challenging tasks in today's era is to deliver personalized information to the user or group of users according to their preferences. The decision making process is used as a tool by the recommender systems to suggest various products and items. The goal of recommender system is to deliver germane information to the user based on their likings. In this research work, we study and compare the effect of similarity measures used in k-means clustering in movie recommender systems. Our proposed method used the sampling, PCA and k-means clustering to recommend movie from the MovieLens dataset. In the whole process, some similarity measures are used in k-means clustering such as Euclidean, Minkowski, Mahalanobis, Cosine similarity and Pearson correlation. The aim of our work is to study the effect of similar measures in movie recommender systems in terms of standard deviation (SD), mean absolute error (MAE), root mean square error (RMSE), t-value, Dunn Matrix, average similarity and computational time using publicly available MovieLens dataset. The results achieved from the experiments indicates that Cosine similarity is best technique in movie recommender system in terms of accuracy, efficiency and processing speed and also able to get MAE of 0.65, which is best between all similarity measures.

Keywords: PCA, K-means clustering, MovieLens dataset, Cosine, Pearson similarity, Minkowski, Mahalanobis

1. Introduction

In the last several years, the use of the internet and its users grows tremendously. The 2nd generation of web i.e. web2.0 is faced great improvements in information and users [16,18]. This web becomes the baseline for information sharing, communication, interoperability, collaboration and user-oriented design [1]. In this generation, main focus shifts from local to global and to distributed systems. Further, many more changes have been found for accessing and creating meaningful information. More and more peoples accessing the information make it very difficult to manage [10,19]. Wikipedia is very good example because so many users access it, modify it and add more contents to it.

The addition of new content increasing per day, make more difficult to find useful data. To overcome this issue, recommender systems are introduced to get the suggestion of the items or data [11,20]. The recommender system becomes very popular now days because it suggests the items according to the likings of the user [15,16]. Recommender system is the information filtering technique used to deal with information overloading by refining important information from the huge information obtained by users and their preferences [2]. Recommender systems can also use to predict whether the given suggestion is according to the user preference or not [12].
A recommender system is used to achieve useful information of the products, information and services to the user by combining preference from the other users, from different authorities and from user attributes [29]. The recommender systems are of two types: collaborative filtering and content-based filtering. The items are selected in collaborative filtering are generally founded on the relation between current users and past users of the product and services. The items are suggested in content-based filtering based on relation between user and its preferences of the product and services [30]. The main difference between both the techniques is that the similarity of preferences between users has been considered in collaborative filtering while in content-based, the similarity of preferences between user and item is considered. Besides these two filtering technique, there is another filtering technique which is the combination of collaborative and content-based filtering.

1.1 Content-based Filtering: Content-based filtering maps the available items with the user’s preferences and items rated in the past to evaluate the best suggestions to the user [31]. This method is different from information retrieval (IR) and information filtering (IF). In information retrieval all the information is retrieved with the help of keywords specify by the user. The information is suggested automatically in the content based filtering. In this technique useful data is stored first either it is taken explicitly or implicitly from the items profile. After analysis of stored data, the system is made suggestion to the user similar to their preferences. Content-based filtering is the subset of the information retrieval [32].

Yih et al. proposed a system which represents items and user’s preferences in the form of vector [33]. Further, these vectors are also used to represent user’s profile and these vectors are processed to get the similarities between items. Arnold et al. introduced a framework which used the information retrieval technique in their content-based filtering called Boolean search indexes [34]. They combine keywords using Boolean operators in their recommendation procedure. Lee et al proposed a framework based on another information retrieval method called probabilistic method. They used the probabilistic method to calculate the probability of the document to check whether it is met with the user’s need or not [35]. Siddiqui et al proposed a system based on natural language processor which can used to retrieve data from natural sentences [36].

Many recommendation systems mostly used content-based filtering to suggest different relevant information to the user. Trivedi et al proposed a system called StubleUpon which was used to suggest user during web browsing. They usually tracked the user likings and browsing history to suggest the appropriate pages to the user [37]. The content-based filtering is also very popular in music recommender systems. Petersen et al. introduced a recommender system to the music world called Last.Fm. It was based on the ratings which are given by the user to the particular music [38]. Liu et al introduced the recommendation of news called google news based on the both collaborative and content-based filtering for the recommendation of the users. They identify user click behaviour to recommend the likings of the articles to the users [39].

1.2 Collaborative-based Filtering: The collaborative-based filtering is based on users with similar likings while content-based filtering is based on likings of similar items. The recommendation for the active users is provided on the basis of similar users and their interests [40]. A group has been formed with the users of same interests. The filtration of information can be done in any source. The relations between user and item are found to be more complex and vast. This feature makes it more powerful as it is able to distinguish between good and bad documents. They are more accurate than other filtering techniques because it takes the benefits of real user rankings instead of machine based rankings [41]. In a music recommendation system, user going to listen classical songs with bad audio quality and user wind up with no more classical songs. The second user also not happy with classical songs but likes few classical songs then this suggestion go to the first user because they both like classical songs. This creates new group and community, which is not possible with the content-based filtering.
The recommender system is useful for both user and service provider [13]. Recommendation systems are also used to decrease the transaction cost for selecting and finding an appropriate item in e-commerce websites [3]. The recommender systems are used to increase the sales of the items to effectively increase the revenue of the organisations [14,17]. In libraries as well the recommender systems predict the user preference more than the catalogue given [4]. Therefore, more efficient, effective and reliable recommendation system concept will give very dependable and accurate predictions for the item [5,6,7,8], which is much needed in the e-commerce applications. The main goals of this work are:

1. To proposed an efficient recommender system using PCA and K-means clustering.
2. Our system is efficient and innovative as it is able to suggest accurate recommendation to the movie using MovieLens dataset.
3. Our proposed work is able to compare different similarity measures used in K-means clustering using standard deviation (SD), mean absolute error (MAE) root mean square error (RMSE), t-value, Dunn Matrix, average similarity and computational time.
4. The cosine similarity has MAE (0.67) in our model which is best among similarity measures.
5. The cosine similarity results are far better than other similarity measure used in our model.

Remaining parts of our paper is organised as: proposed technique will described in section 2, Experiments and results in section 3 and conclusion in section 4.

2. Proposed Methodology

The objective of our work is to study the impact of similarity measures used in K-means clustering in movie recommender system. We use PCA and K-means clustering technique to suggest movie using MovieLens dataset. The freely available dataset used in our proposed method is MovieLens dataset. The data from the MovieLens dataset needs to be pre-processed before using it in machine learning algorithms. Sampling without replacement is used to extract some unwanted data as shown in fig.1. In this method, when item is randomly chosen, it is deleted from population.
PCA is used in already sampled data to reduce dimensionality. PCA is used to produce an ordered list of items that notices the largest variance from mean data in terms of least square errors. Neglecting items having very less contribution to the variance result in dimension reductions.

The k-means clustering is partitioned based technique. The partition of dataset of N into some disjoints subsets Sj so that they are closed as much as possible according to distance measures. Our experiment is typically based on this step where we are going to investigate the impact of similarity measure in k-means clustering in suggesting movies. Each cluster in datasets is divided into clusters Nj with some centroids Cj. Total distance of all data points to centroids should be minimize. The objective of our research is to minimized E from eq. 1 given below.

\[
E = \sum_{j=1}^{k} \sum_{n=1}^{m} d(X_n, C_j)
\]

(1)

Where

\(X_n = \text{n}^{th} \text{ Vector}, C_j = \text{Centroid}, d = \text{distance measure}\)

The interchange of items takes place between clusters until the value of E can’t decreases further.

2.1 Similarity Measures: The most preferred methods used in recommender system is clustering techniques. The clustering techniques are mostly depends on the similarity measures. The details of similarity measures are describe as follows:

**Euclidean distance**: Euclidean distance given by:

\[
d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}
\]

(2)

Where \(n\) represents dimensions and \(x_k\), \(y_k\) are \(k^{th}\) items of data objects \(x\) and \(y\), respectively

**Minkowski Distance**: The distance measure is given by:

\[
d(x,y) = \left( \sum_{k=1}^{n} \left| x_k - y_k \right|^r \right)^{\frac{1}{r}}
\]

(3)

Where \(r\) is the degree of the distance

**Mahalanobis distance**: The distance is given by:

\[
d(x,y) = \sqrt{(x - y)\sigma^{-1}(x - y)^{T}}
\]

Where \(\sigma\) is covariance matrix.

**Cosine Similarity**: Cosine similarity factor is used to calculate the cosine between two non-zero vectors makes an angle \(\theta\) is given by

\[
A \cdot B = \|A\| \cdot \|B\| \cdot \cos (\theta)
\]

(4)
cosine similarity between two vectors is given by using dot product and their magnitude is given by:

\[
\text{Similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \cdot ||\mathbf{B}||}
\] (5)

**Pearson Correlation:** The Pearson correlation is denoted by:

\[
\text{Pearson}(x,y) = \frac{\sum (x,y)}{\sigma_x \sigma_y}
\] (6)

Where \(\sigma\) is standard deviation, \(\sum (x,y)\) is covariance.

### 3. Experiments and Results

We first analyse the MovieLens dataset, which is freely available public dataset. The dataset contains 100000 ratings got from 943 users on 1682 movies on the scale 1 to 5. We proposed a framework which is the combination of PCA and K-means clustering as discussed in section 3. We study the impact of similarity measures in movies recommender systems using MovieLens dataset [9]. We simulate all the experiments in the computer system which has very good configuration of Intel i5 processor, 16 GB RAM and python 3.8.1 environment.

|                  | Euclidean | Minkowski | Mahalanobis | Pearson Correlation | Cosine Similarity |
|------------------|-----------|-----------|-------------|---------------------|------------------|
| SD               | 0.11820   | 0.13324   | 0.12193     | 0.12937             | 0.11353          |
| RMSE             | 1.15983   | 1.17436   | 1.16357     | 1.16947             | 1.14735          |
| MAE              | 0.66      | 0.71      | 0.69        | 0.70                | 0.65             |
| T-value          | 2.67126   | 2.8159    | 2.73854     | 2.79654             | 2.59127          |
| Dunn Index (4 clusters) | 0.35113   | 0.38238   | 0.37453     | 0.36842             | 0.34935          |
| Average Similarity (4 clusters) | 0.95      | 0.93      | 0.96        | 0.94                | 0.98             |
| Computational Time in sec | **61.53** | 78.39     | 105.63      | 94.32               | 83.45            |

To understand the performance of different similarity measures in movie recommender system using MovieLens dataset. We use the metric like standard deviation (SD), mean absolute error (MAE), root mean square error (RMSE), t-value, Dunn Matrix, average similarity and computational time.
In table 1, we compare the results obtained from all similarity measures on the basis of different metrics. The result shows that cosine similarity is among the best distance measures used in k-means clustering technique in movie recommender system using publicly available dataset called MovieLens dataset except in computational time, in which it is better than Pearson and Mahalanobis but slower than Euclidean and Minkowski (see figure 2,3).

In the future, we incorporate some more machine learning concepts like naïve bayes, support vector machine, KNN etc in order to improve the accuracy and processing time and also use it in big data.
References

[1] Sharma, P. (2008). Core characteristics of web 2.0 services. Startup Reviews| Tech news| Tech events| Tech tips| Business Technology Tips Blog Archive.

[2] Pan C, Li W. Research paper recommendations with paper analysis. In computer design and applications IEEE 2010, 4, pp:v4-264.

[3] Pathak B, Garfinkel R, Gopal R, Venkatesan R, Yin F,” Empirical Analysis of Impact of Recommender systems on sale", Journal of Manage Infor System, 2010, 27(2), pp. 159-188.

[4] M Rahul, N Kohli, R Agarwal, S Mishra. Facial Expression Recognition using geometric features and modified hidden markov model, International Journal of Grid and Utility Computing 10 (5), pp. 488-496, 2019.

[5] M Rahul, R Shukla, DK Yadav, V Yadav, Zernike Moment-Based Facial Expression Recognition Using Two-Staged Hidden Markov Model, Advances in Computer Communication and Computational Sciences, pp. 661-670, 2019.

[6] M Rahul, N Kohli, R Agarwal. Facial Expression Recognition using Local Binary Pattern and Modified Hidden Markov Model, January 2018, International Journal of Advanced Intelligence Paradigms, 17(3/4), 2020 DOI: 10.1504/IJAIP.2018.10020898.

[7] M Rahul, R. Shukla, PK Goyal, ZA Siddiqui, V Yadav. Gabor Filter and ICA-Based Facial Expression Recognition Using Two-Layered Hidden Markov Model, In book: Advances in Computational Intelligence and Communication Technology (CICT2019), January 2021, DOI: 10.1007/978-981-15-1275-9_42.

[8] M Rahul, N Kohli, R Agarwal. Facial Expression Recognition using Local Multidirectional Score Pattern (LMSP) descriptor and Modified Hidden Markov Model, January 2018 International Journal of Advanced Intelligence Paradigms 1(1):1, DOI: 10.1504/IJAIP.2018.10020899.

[9] http://grouplens.org/datasets/MovieLens/.

[10] Hug, N., (2020). Surprise: A Python library for recommender systems, Journal of Open Source Software, 5(52), 2174, https://doi.org/10.21105/joss.02174

[11] Emrah Inan, Fatih Tekbacak, Cemalettin Ozturk, Moreopt: A goal programming based movie recommender system, Journal of Computational Science, 28, 2018, pp. 43-50, ISSN 1877-7503, https://doi.org/10.1016/j.jocs.2018.08.004.

[12] Bogdan Walek, Vladimir Fojitik, A hybrid recommender system for recommending relevant movies using an expert system, Expert Systems with Applications, 158, 2020, 113452, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113452.

[13] Enthilselvan Natarajan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, Amir H. Gandomi, Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data, Expert Systems with Applications, 149, 2020, 113248, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113248.
[14] Mohammed Fadhel Aljunid, Manjaiah Dh, An Efficient Deep Learning Approach for Collaborative Filtering Recommender System, *Procedia Computer Science*, 171, 2020, Pages 829-836, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.090.

[15] Bushra Alhijawi, Yousef Kilani, A collaborative filtering recommender system using genetic algorithm, *Information Processing & Management*, 57(6), 2020,102310, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2020.102310.

[16] Sunita Tiwari, Anu Saini, Vaibhav Paliwal, Ajay Singh, Rajat Gupta, Ruchika Mattoo, Implicit Preferences Discovery for Biography Recommender System Using Twitter, *Procedia Computer Science*, 167, 2020, pp. 1411-1420, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.352.

[17] R. Singla, S. Gupta, A. Gupta and D. K. Vishwakarma, "FLEX: A Content Based Movie Recommender", *International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154163.

[18] M. Gupta, A. Thakkar, Aashish, V. Gupta and D. P. S. Rathore, "Movie Recommender System Using Collaborative Filtering," *International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 415-420, doi: 10.1109/ICESC48915.2020.9155879.

[19] R. E. Nakhli, H. Moradi and M. A. Sadeghi, "Movie Recommender System Based on Percentage of View," 2019 5th Conference on *Knowledge Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2019, pp. 656-660, doi: 10.1109/KBEI.2019.8734976.

[20] Y. Deldjoo, M. Schedl and M. Elahi, "Movie Genome Recommender: A Novel Recommender System Based on Multimedia Content," *International Conference on Content-Based Multimedia Indexing (CBMI)*, Dublin, Ireland, 2019, pp. 1-4, doi: 10.1109/CBMI.2019.8877452.

[21] Frias-Martinez, E., Magoulas, G., Chen, S. Y., & Macredie, R. (2006). Automated user modeling for personalized digital libraries, *International Journal of Information Management*, 26(3), 234–248.

[22] Su, X. & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

[23] Ferman, A. M., Errico, J. H., Beek, P. V., & Sezan, M. I. (2002, July). Content-based filtering and personalization using structured metadata. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pp. 393-393, ACM.

[24] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: an introduction. Cambridge University Press.
[25] Yih, W. T. (2009, August). Learning term-weighting functions for similarity measures. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2, pp. 793-802, Association for Computational Linguistics.

[26] Arnold, J. F., & Voss, L. L. (2004). U.S. Patent No. 6, 745, 161. Washington, DC: U.S. Patent and Trademark Office.

[27] Lee, C., & Lee, G. G. (2005). Probabilistic information retrieval model for a dependency structured indexing system, Information processing & management, 41(2), pp. 161-175.

[28] Siddiqui, T. J., & Tiwary, U. S. (2008). Utilizing local context for effective information retrieval, International Journal of Information Technology & Decision Making, 7(01), pp. 5-21.

[29] Trivedi, A., Rai, P., DuVall, S. L., & Daumé III, H. (2010, October), Exploiting tag and word correlations for improved webpage clustering. In Proceedings of the 2nd international workshop on Search and mining user-generated contents, pp. 3-12, ACM.

[30] Petersen, M. K., & Hansen, L. K. (2011, March). Emotional nodes among lines of lyrics. In Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011, pp. 821-826, IEEE.

[31] Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behaviour, In Proceedings of the 15th international conference on intelligent user interfaces, pp. 31-40, ACM.

[32] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE transactions on knowledge and data engineering, 17(6), pp. 734-749.

[33] Ricci, F., Rokach, L., & Shapira, B. (2011), Introduction to recommender systems handbook, pp. 1-35, Springer US.