Movie Recommendation System using Cosine Similarity and KNN

Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, Gaurav Srivastav

Abstract—Over the past years, the internet has broadened the horizon of various domains to interact and share meaningful information. As it is said that everything has its pros and cons therefore, along with the expansion of domain comes information overload and difficulty in extraction of data. To overcome this problem the recommendation system plays a vital role. It is used to enhance the user experience by giving fast and coherent suggestions. This paper describes an approach which offers generalized recommendations to every user, based on movie popularity and/or genre. Content-Based Recommender System is implemented using various deep learning approaches. This paper also gives an insight into problems which are faced in content-based recommendation system and we have made an effort to rectify them.

Keywords: ***Recommendation System, Content-Based Recommender System, Deep learning***

I. INTRODUCTION

Advancement in technology is reaching new heights every day and due to which we can see enormous growth in information. To deal with such large data we use machine learning that automates analytical model building [1]. The early classification of machine learning is divided into three broad categories: Supervised learning, Unsupervised learning and Reinforcement learning [2]. We use computers to make predictions to help us achieve better results using various computational statistics. Tasks can be performed without being explicitly programmed to do so [3]. It becomes a tedious task to extract the relevant information. Search engines solve the problem to some extent but it does not solve the personalization problem. Recommendation System framework plays a vital role in today's internet surfing, be it buying a product from an e-commerce site or watching a movie on some video-on-demand service [4]. In our everyday life, we depend on recommendations given by other people either by word of mouth or reviews of general surveys. People often use recommender systems over the web to make decisions for the items related to their choice. Recommendation systems are software tools and techniques whose goal is to make useful and sensible recommendations to a collection of users for items or products that might interest them [5]. In other words, the recommender system or recommendation systems belongs to a class of information filtering system that aims at predicting the 'preference' or 'rating' given to an item.

Recommendation systems are primarily using three approaches [6]. In content-based filtering, we do profiling based on what type of content any user is interested in and using the collected information, it recommends items. Another one is collaborative filtering, where we make clusters of similar users and use that information to make recommendations. Hybrid systems are the one which takes into account both above stated approaches to deal with operational data more concisely [7]. Our goal is to provide accurate recommendations with less computational complexity.

II. RELATED WORK

Some of the common approaches of recommender system are:

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

A. Content Based Filtering

This approach filters the items based on the likings of the user. It gives result based on what the user has rated earlier. The method to model this approach is the Vector Space Model (VSM). It derives the similarity of the item from its description and introduces the concept of TF-IDF (Term Frequency-Inverse Document Frequency) [28].

$$\text{Tf(t)}=\frac{\text{frequency occurrence of term t in document}}{\text{totalnumberoftermsindocumen } t}$$

$$\text{If(t)}= log10\frac{\text{totalnumberofdocument}}{\text{numberofdocumentscontainingtermt}}$$

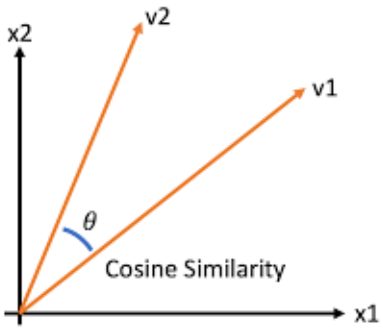
The similarity between item vectors can be computed by three methods:

1. Cosine similarity
2. Euclidian distance
3. Pearson's correlation

COSINE SIMILARITY

Cosine similarity among two objects measures the angle of cosine between the two objects. It compares two documents on a normalized scale. It can be done by finding the dot product between the two identities.

Revised Manuscript Received on May 29.2020.
Gaurav Srivastav, Inderprastha Engineering College, AKTU
Ramni Harbir Singh, Inderprastha Engineering College, AKTU
Sargam Maurya, Inderprastha Engineering College, AKTU
Tanisha Tripathi, Inderprastha Engineering College, AKTU **Tushar Narula,** Inderprastha Engineering College, AKTU



[25]

As the above diagram shows, the angle between v1 and v2 is. Lesser the angle between the two vectors more is the similarity. It means if the angle between two vectors is small, they are almost alike each other and if the angle between the two vectors is large then the vectors are very different from each other.

B. Collaborative Filtering

It depends upon the users who have similar interests and gives the result based on all the users.

User-based: In user-based collaborative filtering, it is considered that a user will like the items that are liked by users with whom have comparable taste.

Item-based: Item-based collaborative-filtering is different, it expects the users to like items that are related to items that he has liked earlier.

C. Hybrid Filtering

Hybrid filtering can be known as a combination of collaborative filtering and content-based filtering. It is the most common and popular technique today. It avoids the weakness of every single recommender technique.

There are some problems related to the recommendation system. They are:

Cold-start problem: When a user registers for the first time, he has not watched any movie. So, the recommendation system does not have any movie based on which it can give results [8]. This is called the cold-start problem [9]. Recommendation system goes through this problem as a result of no previous record. This happens with every new user once.

Data sparsity problem: This problem occurs when the user has rated very few items based on which it is difficult for the recommendation system to give accurate results. In this problem, the results given are not much similar to the expected result. Sometimes, the system fails to give successful results and generates weak recommendations [10]. Also, data sparsity always leads to coverage problems.

Scalability: Another problem associated with recommendation is scalability. In this, the encoding goes linearly on items. The system works efficiently when the data set is of limited size [11]. As the data set increases, it becomes difficult for the recommendation system to give accurate results based on varying genres of movies. The scalability problem can be solved by dimensionality reduction technique such as singular value decomposition (SVD). It also speeds up the generation of result and produces a reliable and efficient result.

Synonym: When many words have a similar meaning then they are known as synonyms. In this problem, the system sometimes fails to understand the difference between two similar words and cannot produce the desired output. E.g.: movie and film have the same meaning but the recommendation system considers them as different words and because of this it fails to give the accurate output. Singular Value Decomposition (SVD), especially Latent Semantic Indexing is capable of solving the synonymy problem [12]. In other recommendation systems, recommendations are based on the ratings and genres given by other users due to which it became difficult to give accurate results. This is called collaborative filtering. This is a common method for a recommendation system. To make things easier for people and come over this drawback, we have used another method for the recommendation which is content-based filtering. This method uses ratings and genres given by the user itself. It gives recommendations based on the movies watched by the user earlier. This helps us in giving accurate suggestions because the ratings and genres are given by the user only and depend entirely on the user's choice, not on any other person's choice. Previous researches lack the accuracy of true recommendations due to their dependency on other users. Here it is important to mention that recommendations can be based on a particular user and they assure to give recommendations on the mark as they depend on the likes and dislikes of a particular user. The proposed system is capable of storing a large amount of data and give efficient results. Our recommendation system searches for the best movies that are similar to the movie that we have watched based on genre and gives the result.

III. RESULT AND DISCUSSION

In the deep learning framework recommendation system, we have used Cosine Similarity and Content-based filtering to predict our result and recommend a movie to the user by running the code in python using NumPy and panda libraries.

A. Experiment Result

The formula used to measure how similar the movies are based on their similarities of different properties. Mathematically, it shows the cosine of the angle of two vectors projected in a multidimensional space. The cosine similarity is very beneficial since it helps in finding similar objects.

$$CosSim(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

Fig: This is the Cosine similarity formula which is used for the recommendation of movies

Fig: To implement Cosine similarity we take an example of 2 movies of different genre adventure and comedy

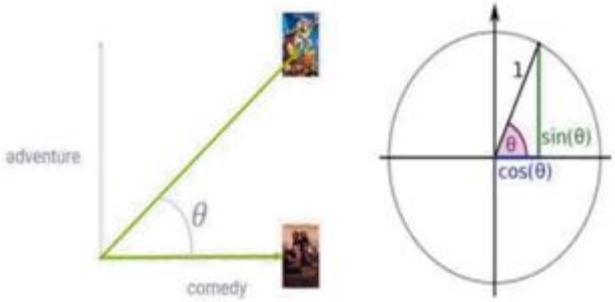


Fig: Cosine similarity

The angle theta between the two movies will determine the similarity between the two movies. The theta ranges from 0-1. If the value of the theta is near 1 then it is most similar and if it's near to 0 then it is least similar. The movie will be recommended if it is close to 1 otherwise there would be no similarity between them. It will recommend the best movies to the user according to the Cosine similarity. After the cosine similarity, we have used a normalised popular score through which we get our function of computing distance. Then by using the KNN functionality, we have found the nearest neighbour which will be recommended to the user.

movield	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

Fig: Movies with different genres are takes and their similarity is taken with the help of cosine similarity

IV. CONCLUSION

We have illustrated the modelling of a movie recommendation system by making the use of content-based filtering in the movie recommendation system. The KNN algorithm is implemented in this model along with the principle of cosine similarity as it gives more accuracy than the other distance metrics and the complexity is comparatively low too

Recommendations systems have become the most essential fount of a relevant and reliable source of information in the world of internet. Simple ones consider one or a few parameters while the more complex ones make use of more parameters to filter the results and make it more user friendly. With the inclusion of advanced deep learning and other filtering techniques like collaborative filtering and hybrid filtering a strong movie recommendation system can be built. This can be a major step towards the further development of this model as it will not only become more efficient to use but also increase the business value even further.

REFERENCES

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.
2. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM Press New York, 1999
3. ShumeetBaluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In Proceedings of the 17th international conference on World Wide Web, pages 895–904. ACM, 2008.
4. Xu Hailing, Wu Xiao, Li Xiaodong, and Yan Baoping. Comparison study of internet recommendation system. Journal of Software, 20(2):350–362, 2009.
5. T. E. D. Mining, “Enhancing teaching and learning through educational data mining and learning analytics: An issue brief,” in Proceedings of a conference on advanced technology for education, 2012.
6. Nakagawa and T. Ito, “An implementation of a knowledge recommendation system based on similarity among users’ profiles,” in Sice 2002. roceedings of the Sice Conference, 2002, pp. 326–327 vol.1.
7. T. K. Quan, I. Fuyuki, and H. Shinichi, “Improving the accuracy of recommender system by clustering items based on the stability of user similarity,” in International Conference on Computational Intelligence for Modelling Control and Automation, 2006, p. 61.

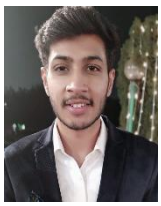
8. M. Muozorganero, G. A. Ramezgonzlez, P. J. Muozmerino, and C. D Kloos, “A collaborative recommender system based on space-time similarities,” vol. 9, no. 3, pp. 81–87, 2010.
9. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.
10. G. Wang, “Survey of personalized recommendation system,” Computer Engineering & Applications, 2012.
11. Albadvi and M. Shahbazi, “A hybrid recommendation technique based on product category attributes,” Expert Systems with Applications, vol. 36, no. 9, pp. 11 480–11 488, 2009.
12. F. R. Hernandez and N. Y. G. Garcia, “Distributed processing using cosine similarity for mapping big data in Hadoop,” IEEE Latin America Transactions, vol. 14, no. 6, pp. 2857–2861, 2016.
13. Anant Gupta, Dr.B.K.Tripathy.A generic hybrid recommender system based on neural networks. Advance Computing Conference (IACC), 2014 IEEE International. 21-22 February 2014.
14. Graham L. Giller (2012). ”The Statistical Properties of Random Bit-streams and the Sampling Distribution of Cosine Similarity”. Giller Investments Research Notes (20121024/1).
15. Nandhini Rengaraj. C.M.Kavitha. .Sabitha Loganathan. N.Muthurasu. A study of existing systems of recommendation engines. National Conference of Emerging Computing Technologies and Applications. Vel Tech University. Avadi, Chennai. 05-06 April 2018.
16. Robert Bell, Yehuda Koren, and Chris Volinsky. Modelling relationships at multiple scales to improve the accuracy of large recommender systems. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 95–104. ACM, 2007.
17. Suvir Bhargav. Efficient features for movie recommendation systems. 2014.
18. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: an introduction. Cambridge University Press, 2010.
19. Joseph A Konstan. Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems (TOIS), 22(1):1–4, 2004.
20. Vilakone, P., Park, D., Xinchang, K. et al. An Efficient movie recommendation algorithmbased on improved k-clique. Hum. Cent. Comput. Inf. Sci. 8, 38 (2018).
21. Kelvin Luk, Introduction to TWO approaches of content-based recommendation system
22. https://en.wikipedia.org/wiki/Recommender_system
23. https://medium.com/@bkexcel2014/building-movie-recommender-systems-using-cosine-similarity-in-python-eff2d4e60d24
24. https://www.kaggle.com/rounakbanik/movie-recommender-systems
25. https://towardsdatascience.com/movie-recommender-system-part-1-7f126d2f90e2
26. https://campus.datacamp.com/courses/introduction-to-natural-language-processing-in-r/representations-of-text?ex=12
27. https://www.researchgate.net/publication/11563559_Comparative_methods_for_the_analysis_of_continuous_variables_Geometric_interpretations
28. https://towardsdatascience.com/calculate-similarity-the-most-relevant-metrics-in-a-nutshell-9a43564f533e?gi=79783aadb374
29. https://www.ijcaonline.org/archives/volume124/number3/22082-2015904111
30. https://www.forbes.com/sites/mitsubishiheavyindustries/2019/01/23/how-hydrogen-fuel-cells-can-power-the-world/#19e6cdb05695
31. https://github.com/jeknov/movieRec/blob/master/movieRec_descr.Rmd
32. https://www.sciencedirect.com/science/article/pii/S000437021500137X
33. https://www.researchgate.net/publication/266655527_Choice-Based_Preference_Elicitation_for_Collaborative_Filtering_Recommender_Systems

AUTHORS PROFILE

Ramni harbir singh. Final year student of computer science engineering in Inderprastha Engineering College, batch of 2016-20. Completed internship in core java from CEBS, Noida. Also has certification in python.

Sargam Maurya, final year student of computer science in Inderprastha Engineering College of APJ Abdul Kalam Technical University batch 2016-20. She has a certified training of Ethical Hacking from internshala and Python from coding ninjas.

Tanisha Tripathi, final year student of computer science in Inderprastha Engineering College of APJ Abdul Kalam Technical University batch 2016-20. She has completed her internship in android from Ducat, Noida and also has a certification in python.



Tushar Narula, final year student of computer science in Inderprastha Engineering College of APJ Abdul Kalam Technical University batch 2016-20. He has done training in Java and Web Development from Webtek Labs Pvt. Ltd and training in Python from coding blocks.

Gaurav Srivastav is working as an Asst. Professor in Inderprastha Engineering College,AKTU. He has completed his B. Tech, M. Tech, and currently pursuing PHD. He has worked for Ericsson and Nokia in RF department. His area of expertise is machine learning, e-learning.