Movie Recommender System Using Genetic AlgorithmPaper

Subhankar Joardar[1], Pinaki Pratim Acharjya[2], Sanchita Saha[3], Sumanta Kuila[4]
1,2,3,4(Department of ComputerScience & Engineering, Haldia Institute of Technology,India)

ABSTRACT
A movie recommendation is important in our social life due to its strength in providing spectacularly improvised entertainment. Such a system can suggest a set of movies to users based on their interest of the movies. A set of movie recommendation systems have been proposed In this project we have explored and implemented various movie recommendation systems that have the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. The implemented systems prove effectiveness of the proposed systems.

Keywords – SVM, LSVM, Genetic Algorithm, Cosine filtering, Collaborative Filtering

1. INTRODUCTION

The recommendation system widely used in the routine life where people rely on knowledge for deciding their interests. The collaborative filtering model takes data from a user's previous behavior (i.e., previously purchased items or chose or numerical ratings provided to the items) as well as similar decisions made by other users. After that, different models are used to forecast items (or ratings for items) that the user might have an interest in. Although there are many approached developed in the past. However, search still goes on due to its often used in many applications, which personalize recommendation and deal with a lack of accuracy. These demands throw some challenges. To solve this, many researchers have used algorithms like Alternating Least Squares, Singular Value decomposition, K-Nearest Neighbor algorithm, and Normal predictor algorithm. Collaborative filtering techniques divided into memory-based and model-based methods. Memory-based methods take action only on a user-item rating matrix and can easily be adjusted to use all the ratings before the filtering procedure; thus, its results updated.

On the other hand, a model-based system, like a neural network, generates a model that learns from the information of user-item ratings and recommends new items — following shows the detailed description of all the above approaches. The recommender system still requires improvement to develop a better and accurate method. The recommendation system is a sharp system that provides ideas about the item to users that might interest them. In this paper, different approached.

The evolution of technology brings us many advanced platforms such as Machine Learning, Deep Learning, Data Mining, the Internet of Things (IoT), etc. To satisfy the need of society, almost in each work, we use this technology. It has many real-life applications such as PowerShell [1], TP [2-4], IoT [5-12], Cloud Computing [13], Artificial Intelligence [14], Uncertainty [15-17], virtualization Environment [18], SPP [19-26], and so on. IT is the mode to store, fetch, communicate and utilize the information. So, all the organizations, industries and also every individual are using computer systems to preserve and share the information. As we probably are aware, the world is becoming quicker and everybody is moving towards accomplishing their objectives. Individuals need more time to go to the market and purchase things, not simply that, they don't have the opportunity to pick between things. What's more, this has prompted the innovation of recommendation systems [27, 28]. Recommendation systems have become well.

Known nowadays, be it in the field of entertainment, education, etc. Earlier, the users needed to settle on choices on what books to purchase, what music to tune in to, what motion pictures to watch and so on. Commercial movie libraries effectively exceed 15 million films, which boundlessly exceeds the visual ability of any single individual. With a large number of motion pictures to browse, individuals now and then get overpowered. Therefore, an efficient recommendation system is necessary for the enthusiasm of both movie service providers and customers [29]. With the improvement of recommendation systems, the customers will have no agony in settling on choices and organizations can keep up their client gathering and draw in new clients by improving users' satisfaction [30, 31]. Additionally, nowadays the modern technologies like machine learning and deep learning also plays a vital role in the process flexible technologies for day to day operations. In this manuscript, we discuss about the recommendation by using machine learning. Now, we discuss a method that has been previously implemented

1.1 Rating Based Movie Recommendation System

1.1.1 Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

1.1.2 Linear SVM

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

1.2 Genre based movie recommender system

The approach used for building the recommendation system is Genre-based filtering.Genre- based filtering analyses user's choice of movie and recommends items similar to it based on the parameters considered. This aims at recommending movies to users based on similarity of genres.

1.2.1 Cosine similarity method

Recommendation Systems work based on the similarity between either the content or the users who access the content.There are several ways to measure the similarity between two items. The recommendation systems use this similarity matrix to recommend the next most similar product to the user.

In this system, we will build a machine learning algorithm that would recommend movies based on a movie the user likes. This Machine Learning model would be based on Cosine Similarity.

Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1. 0 means no similarity, whereas 1 means that both the items are 100% similar.

Cosine similarity , $$\cos\theta = \frac{A.B}{||A||*||B||} \quad (1)$$

1.3 Collaborative filtering

To address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features.

Therefore it is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions. There are many ways to decide which users are similar and combine their choices to create a list of recommendations.

1.3.1 Memory based collaborative filtering

The researchers came up with a reduction technique to store only the relevant data according to the distance (or similarity) based neighborhood approach on the raw data, so that the further calculations can be done on the user's preferences alone. This technique is called Dimensionality Reduction.

Reasons to reduce dimensionality:

- ✓ Computational issues with large number of predictors
- ✓ Certain statistical methods like regression could not be applied when the density of observations per predictor is low.
- ✓ Noisy data provides less accurate results

In other words, only the tastes and preferences of the user is derived from the raw data using the Dimensionality Reduction aka, low-rank Matrix factorization.

1.4 Genetic Algorithm

There has been an explosion, in recent years, of the number of websites available on the internet that recommend movies. This can result in the user being inundated by a great number of websites to view, highlighting the subject matter and providing reams of recommendation, most of which will be irrelevant for the users' needs, due to drawbacks in the recommendation systems. To solve this problem, the Genetic algorithm based collaborative recommender system has been developed to filter the movies by choosing only the most appropriate movies for the user.

Although there are many recommendation algorithms developed to improve the performance of the Recommendation System, they still need to overcome the cold-start and sparsity problems. The cold-start problem occurs when the Recommendation System does not have enough information about a new user and/or a new item. The sparsity problem occurs when the frequency of the purchased items is too small. These problems have been overcome using the Genetic algorithm.

II. MOVIE RECOMENDER SYSTEM USING GENETIC ALGORITHM

As a heuristic search algorithm that mimics the process of natural evolution, the genetic algorithm has been widely applied to many applications in different Recommendation Systems. In the process of the Genetic Algorithm, there are five parts: the initial population, evaluation, reproduction, crossover operation, and mutation operation. The population of the Genetic Algorithm is a group of chromosomes consisting of genes (an array of values). By mimicking natural selection, the chromosomes with a high fitness value are selected into a mating pool. The reproduction process occurs in the pool by copying individual chromosomes to the next generation. The crossover operation creates children from the parents based on the pairing process. The mutation operation aims to maintain genetic diversity from one generation of the population to the next.

A Fitness Score is given to each individual which shows the ability of an individual to "compete". The individual having optimal fitness score (or near optimal) are sought.

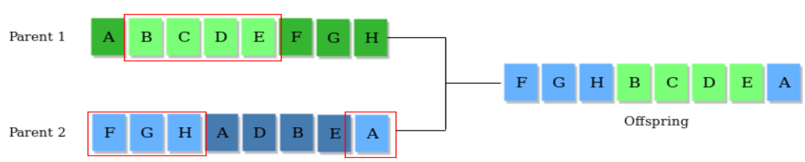
The Genetic Algorithms maintains the population of n individuals (chromosome/solutions) along with their fitness scores. The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce better offspring by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

Each new generation has on average more "better genes" than the individual (solution) of previous generations. Thus each new generations have better "partial solutions" than previous generations. Once the offsprings produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

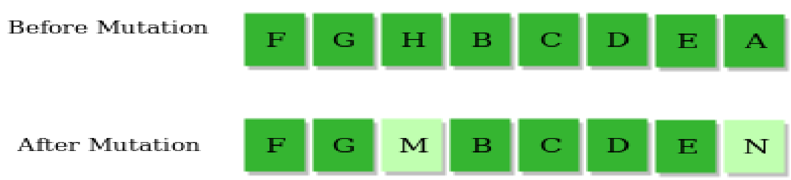
Operators of Genetic Algorithms:

Once the initial generation is created, the algorithm evolve the generation using following operators –

- ✓ **Selection Operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass there genes to the successive generations.
- ✓ **Crossover Operator:** This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring).
- ✓ **Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in



population to avoid the premature convergence.

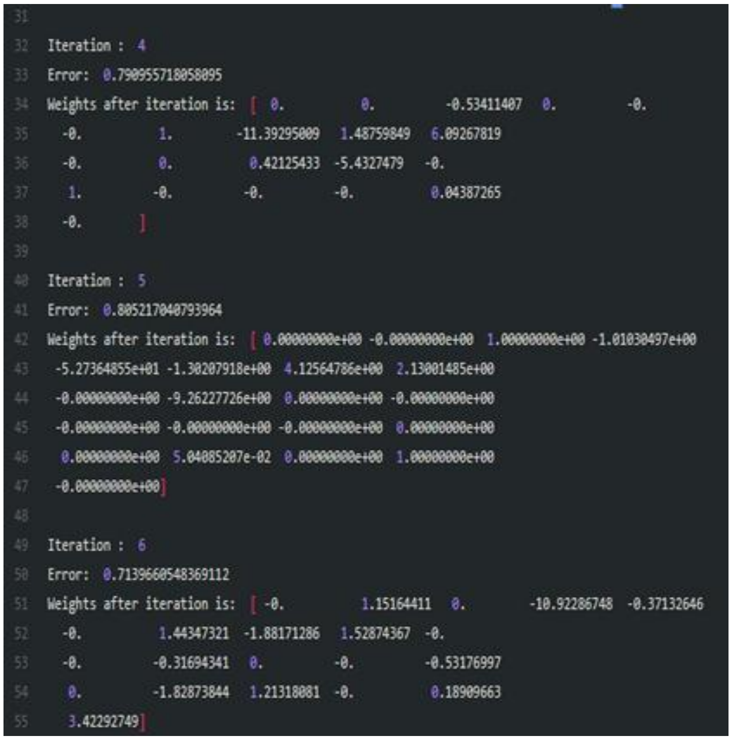


Pseudo code for implementation of Genetic Algorithm :

1. Randomly initialize populations p
2. Determine fitness of population
3. Untill convergence repeat:
    a. Select parents from population
    b. Crossover and generate new population
    c. Perform mutation on new population
    d. Calculate fitness for new population

IV. OUTPUT

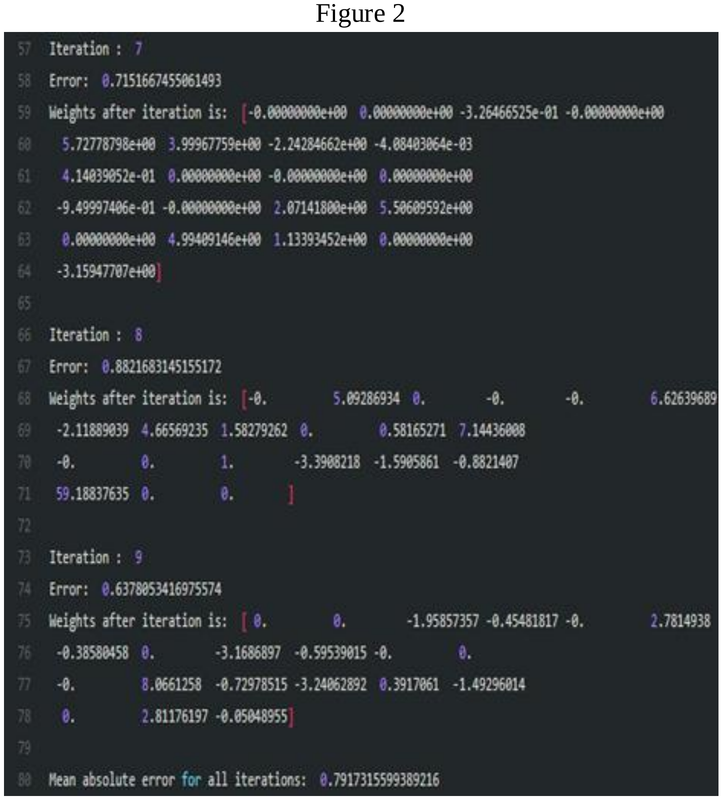
Figure -1


```
Iteration :  4
Error:  0.790955718058095
Weights after iteration is:  [ 0.          0.          -0.53411407  0.          -0.
  -0.          1.          -11.39295009  1.48759849  6.09267819
  -0.          0.          0.42125433  -5.4327479  -0.
  1.          -0.          -0.          -0.          0.04387265
  -0.        ]

Iteration :  5
Error:  0.805217040793964
Weights after iteration is:  [ 0.00000000e+00 -0.00000000e+00  1.00000000e+00 -1.01030497e+00
 -5.27364855e+01 -1.30207918e+00  4.12564786e+00  2.13001485e+00
 -0.00000000e+00 -9.26227726e+00  0.00000000e+00 -0.00000000e+00
 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00  0.00000000e+00
  0.00000000e+00  5.04085207e-02  0.00000000e+00  1.00000000e+00
 -0.00000000e+00]

Iteration :  6
Error:  0.7139660548369112
Weights after iteration is:  [ -0.          1.15164411  0.          -10.92286748  -0.37132646
  -0.          1.44347321  -1.88171286  1.52874367  -0.
  -0.          -0.31694341  0.          -0.          -0.53176997
  0.          -1.82873844  1.21318081  -0.          0.18909663
  3.42292749]
```

Figure 2


```
Iteration :  7
Error:  0.7151667455061493
Weights after iteration is:  [-0.00000000e+00  0.00000000e+00 -3.26466525e-01 -0.00000000e+00
  5.72778798e+00  3.99967759e+00 -2.24284662e+00 -4.08403064e-03
  4.14039052e-01  0.00000000e+00 -0.00000000e+00  0.00000000e+00
 -9.49997406e-01 -0.00000000e+00  2.07141800e+00  5.50609592e+00
  0.00000000e+00  4.99409146e+00  1.13393452e+00  0.00000000e+00
 -3.15947707e+00]

Iteration :  8
Error:  0.8821683145155172
Weights after iteration is:  [-0.          5.09286934  0.          -0.          -0.          6.62639689
 -2.11889039  4.66569235  1.58279262  0.          0.58165271  7.14436008
 -0.          0.          1.          -3.3908218  -1.5905861  -0.8821407
 59.18837635  0.          0.        ]

Iteration :  9
Error:  0.6378053416975574
Weights after iteration is:  [ 0.          0.          -1.95857357 -0.45481817 -0.          2.7814938
 -0.38580458  0.          -3.1686897  -0.59539015 -0.          0.
 -0.          8.0661258  -0.72978515 -3.24062892  0.3917061  -1.49296014
  0.          2.81176197 -0.05048955]

Mean absolute error for all iterations:  0.7917315599389216
```

Figure 3

V. CONCLUSION

Using the Genetic algorithm, each new generation have better partial solutions than the previous one. Once the offsprings produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converted to a set of solutions for the problem extensions. After total no of nine iterations we have found a handsome good result as .7917 % accuracy that is very close to 80% accuracy.

REFERENCES

1. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012
2. Cui, Bei-Bei. (2017). Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm. ITM Web of Conferences. 12. 04008. 10.1051/itmconf/20171204008.
3. Fadhel Aljunid, Mohammed & D H, Manjaiah. (2018). Movie Recommender System Based on Collaborative Filtering Using Apache Spark. https://doi.org/10.1007/978-981-13-1274-8_22
4. Miryala, Goutham & Gomes, Rahul & Dayananda, Karanam. (2017). COMPARATIVE ANALYSIS OF MOVIE RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING IN SPARK ENGINE. Journal of Global Research in Computer Science. 8. 10-14.
5. Banerjee, Anurag & Basu, Tanmay. (2018). Yet Another Weighting Scheme for Collaborative Filtering Towards Effective Movie Recommendation.
6. Zhao, Zhi-Dan & Shang, Ming Sheng. (2010). User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. 3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010. 478-481. https://doi.org/10.1109/WKDD.2010.54
7. Kharita, M. K., Kumar, A., & Singh, P. (2018). Item-Based Collaborative Filtering in Movie Recommendation in Real-time. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). DOI:10.1109/icsccc.2018.8703362
8. A. V. Dev and A. Mohan, "Recommendation system for big data applications based on the set similarity of user preferences," 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, 2016, pp. 1-6. DOI: 10.1109/ICNGIS.2016.7854058
9. Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A. and Vijayakumar, V. (2017) ‘A personalized movie recommendation system based on collaborative filtering,’ Int. J. High-Performance Computing and Networking, Vol. 10, Nos. 1/2, pp.54–63.
10. Thakkar, Priyank & Varma (Thakkar), Krunal & Ukani, Vijay & Mankad, Sapan & Tanwar, Sudeep. (2019). Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning: Proceedings of ICTIS 2018, Volume 2. 10.1007/978-981-13-1747-7_17.