Comparative Analysis of Clustering Techniques for Movie Recommendation

Aditya TS[1], *Karthik* Rajaraman[2], and *M. Monica* Subashini[3,*]

[1]School of Electronics Engineering, VIT University, Vellore, India.
[2]School of Computer Science Engineering, VIT University, Vellore, India
[3]School of Electrical Engineering, VIT University, Vellore, India

Abstract. Movie recommendation is a subject with immense ambiguity. A person might like a movie but not a very similar movie. The present recommending systems focus more on just few parameters such as Director, cast and genre. A lot of Power intensive methods such as Deep Convolutional Neural Network (CNN) has been used which demands the use of Graphics processors that require more energy. We try to accomplish the same task using lesser Energy consuming algorithms such as clustering techniques. In this paper, we try to create a more generalized list of similar movies in order to provide the user with more variety of movies which he/she might like, using clustering algorithms. We will compare how choosing different parameters and number of features affect the cluster's content. Also, compare how different algorithms such as K-mean, Hierarchical, Birch and mean shift clustering algorithms give a varied result and conclude which method will suit for which scenarios of movie recommendations. We also conclude on which algorithm clusters stray data points more efficiently and how different algorithms provide different advantages and disadvantages.

1 Introduction

A movie recommendation system using four different clustering algorithms is built on the same cleaned dataset with identical features. The clusters are then compared and the most desirable algorithm is pointed out. The python library scikit-learn is used to implement the same. In-built clustering algorithm functions are called and processed in the datasets. Depending on number of clusters wanted, and other internal features the clustering again changes.

1.1 Literature review

A commonly used unsupervised learning methodology is Clustering. Clustering is essentially grouping the set of instances in a way that those within the same group (cluster) are similar to each other than to those in the other clusters.

* Corresponding author: monicasubashini.m@vit.ac.in

A cluster therefore is defined as a collection of samples which are similar between them and are dissimilar to the samples belonging to other clusters. In reference [1], the use of clustering algorithm to surf through large data in order to search, scatter or gather them is vividly discussed. [2] Similarly discusses about the usage of hierarchical agglomerative clustering algorithms mainly for document retrieval. These can show us the idea of usage of clustering techniques and the scope of applications it holds. [4] Discusses how conventional distance based clustering algorithms are not suitable for Boolean based features and hence propose another algorithm to do the same. Reference [11] discusses Energy efficiency of running intensive CNN such as Deep CNN on CPUs and GPUs. It observes how simple clustering algorithms require lesser Energy than for Deep Learning Algorithms.

1.2 Proposed method

We have used Euclidean distance based algorithms like K-means and have compared with clustering algorithms such as Agglomerative, Birch and Mean-shift techniques. The datasets are pre-processed, cleaned and fed into the algorithms to cluster them based on specific features. Selective features of the datasets are used and the significance in the similarities of the movies in cluster are studied. Here, no user fed input is compared with the output of the clusters just because of the ambiguity of the subject as discussed earlier in abstract.

1.3 Significance and novelty

Using the Agglomerative, Birch and Mean-shift techniques for movie recommendation is a novel approach used for this application. The variety of feature combination used, introduces the user to a more variety of movies instead of restricting to specific set of similar movies. Similarity in movies are recognized on different aspects instead of focusing on specific attributes such as cast, director and genre. In this way, clustering is occurring in a broader perspective but yet with a good similarity within each other.

2 Clustering algorithms

2.1 Agglomerative clustering

The Agglomerative hierarchical clustering is a bottom-up clustering method wherein clusters have sub-clusters, which in turn have sub-clusters and so on where a traditional example of this could be the species taxonomy. Agglomerative hierarchical clustering begins with every single instance in a single cluster. The process involves construction of a distance matrix and combining the pair of clusters that have the shortest distance. Usage of different distance metrics for the purpose of measuring distances between clusters may give different results. Performing multiple experiments and comparing the results is recommended to support the veracity of the original results.

2.2 Birch clustering

The Birch algorithm has two specifications, namely- the threshold and the branching factor. The branching factor limits the number of sub-clusters in a node. The threshold factor limits the distance between the entering sample and the existing sub-clusters. Birch Clustering involves building a tree which is referred to as the Characteristic Feature Tree (CFT). The dataset is eventually compressed into a set of Characteristic Feature nodes denoted as CF

Nodes. These CF Nodes have a number of sub-clusters - denoted by CF Sub-clusters. These CF Sub-clusters located in the non-terminal CF Nodes, can have CF Nodes as children

2.3 K-means clustering

The K-Means algorithm minimizes a criterion given as the sum of squares of intracluster distances and hence it clusters the data by distinguishing the samples into n groups of equal variance. This algorithm requires the number of clusters to be specified. K-means has been widely used various fields and across different applications owing to its capability to scale well to large datasets.

2.4 Mean shift clustering

Mean Shift is non parametric iterative algorithm that is versatile and hence can be used for lot of purposes like finding modes, clustering etc. Mean shift is a procedure for finding the maxima of density function that is generated by the discrete sample data. We define a kernel function that determines the weights of samples for the iterative estimation of mean. The difference between the weighted mean and each of the sample is known as the mean shift and the algorithms repeats the estimation un-till this value converges.

3 Methodology

3.1 Introduction

Dataset: TMDB 5000 Movie Dataset by “The Movie Database” on kaggle.com
Pre-processing: Cleaning/Pruning of the dataset: We loaded the movies dataset and removed all the fields which have irrelevant or invalid values (NaN). Then we also select the dominant features from the dataset that we wished to focus on. This enabled our study to be more efficient and avoid unwanted outputs in our final result. All the features were encoded into numbers so as to help the clustering process. We had to select two features to compare and cluster them into similar movies.

3.2 Implementation

Python libraries sci-kit learn, matplotlib and pandas were used to code the clustering programs. Pre-processing, reading and cleaning of the datasets was done using pandas library. Inbuilt classes and function of sklearn was used to implement the different clustering techniques. Appropriate parameters and input features were plugged in to the functions of the clusters. Matplotlib helped us to provide appropriate scatter plots with the different feature combinations plot.

4 Clustering results

The graphs are numbered 1 to 4 clockwise starting from title year vs IMDB score chart.

4.1 Agglomerative clustering

The first graph gives a spread clustered result. It has taken a wide coverage of movies based on the features selected. The other clusters was been able to see when the color coding was

changed. While, the second combination gave a more layered output and depended more on the popularity of cast in social media. The third combination didn't show desired results and gave poor clustering. From 4th and 2nd combinations, we can observe how the title year and the IMDB score shifted the clusters differently

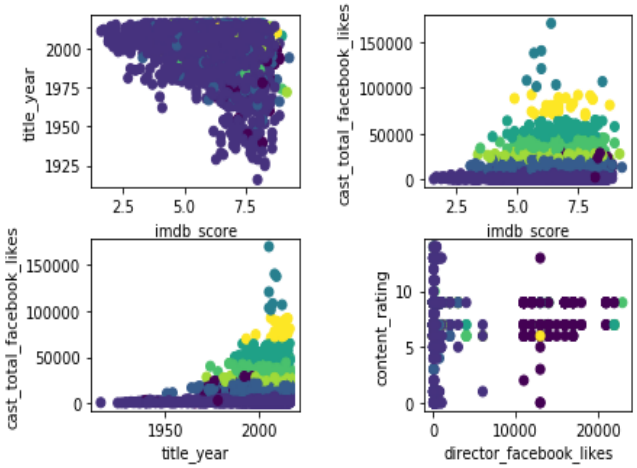

Fig. 1. Agglomerative clustering results

4.2 Birch clustering

The Balanced Iterative Reducing and Clustering using Hierarchies- BIRCH clustering provided similar results to the former with a distinctive difference in the 2nd graph; the clustering has taken a wider coverage area on the basis of the total Facebook likes of cast. In that manner, the results gave more variety of movies in one cluster itself. Similarly for the 3rd graph, clustering was found to be more 'thicker'. The 4th graph did not give satisfactory results and was not taken into consideration.

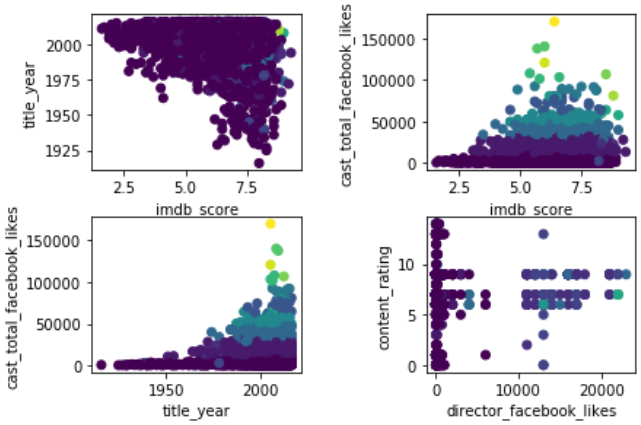

Fig. 2. Birch clustering results

4.3 K-means clustering

This technique handled the stray data points in a more efficient way. It was able to group it to the most similar cluster. The clustering used here was more fine-tuned with 'thinner' cluster with respect to the total Facebook likes of cast and had used more number of clusters. The 4th graph again had undesirable results due to the ineffective selection of feature combinations.

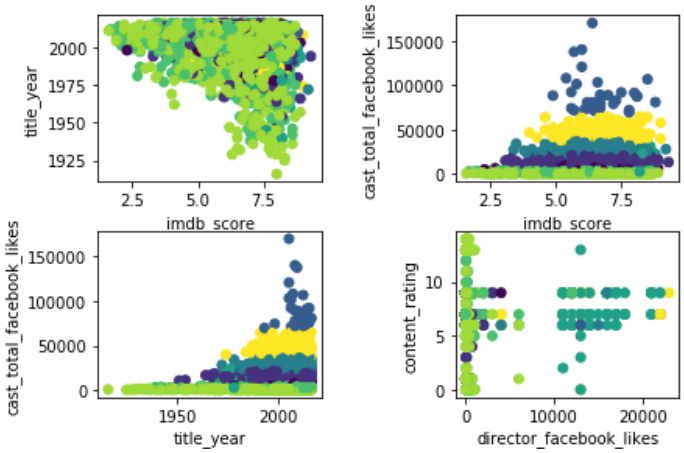

Fig. 3. K-means clustering results

4.4 Mean shift clustering

This produced the lowest quality of clusters with respect to the other algorithms in play. The stray data points were handled as different clusters altogether and hence providing a separate cluster with just 2 or 3 data points. Since in mean shift clustering, there was no need to define the number of clusters, the algorithm tended to take freedom to create more number of clusters in-turn giving sparsely populated clusters.

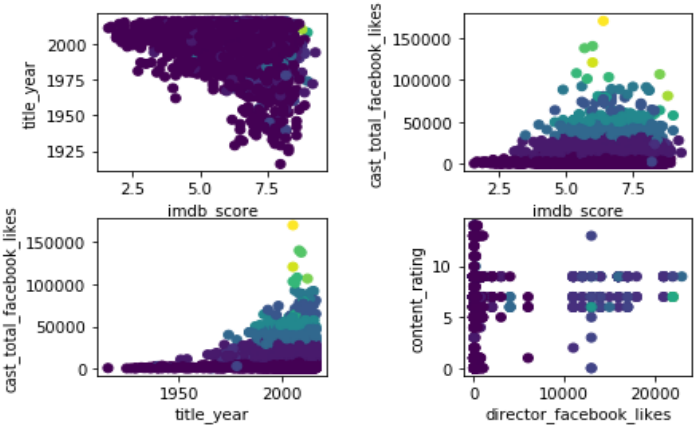

Fig. 4. Mean shift clustering results

4 Inferences

Table 1. Some of the clustered data after K-means clustering

Cluster ID	ID	Movie Title
Cluster 0	11	Superman Returns
	32	Iron Man 3
	46	World War Z
	50	The Great Gatsby
	69	Iron Man
Cluster 1	1	Pirates of the Caribbean: At World's End
	6	Spider-Man 3
	9	Harry Potter and the Half-Blood Prince
	13	Pirates of the Carribean: Dead Man's Chest
	14	The Lone Ranger
Cluster 2	3	The Dark Knight Rises
	8	Avengers: Age of Ultron
	17	The Avengers
	33	Alice in Wonderland
	38	Oz the Great and Powerful
Cluster 3	5	John Carter
	7	Tangled
	12	Quantum of Solace
	30	Skyfall
	49	Jack the Giant Slayer

Table 1 shows a sample of five movie titles from each cluster formed by K-Means clustering. It can be observed that action movies such as "Iron Man" and "World War Z" have been clustered together (cluster 0) and superhero movies such as Batman, Man of Steel and the Amazing Spider-Man were clustered together. (cluster 5).

This paper depicts the results of experimenting and analyzing the effects of the common clustering techniques on the selected dataset of movies. The scenario of usage- for movie recommendation system proved to give a better scope of usage for unsupervised learning as the liking of a movie to oneself is ambiguous and may vary from person to person. Some movies (stray data points) which were totally different from the major clusters were handled differently by different algorithms and was analyzed briefly. The feature combinations gave different outputs and were able to obtain similar movies depending on those attributes which the user could choose from.

Inferences obtained:
• Achieved good similarity results without using Deep CNN and other Energy intensive algorithms by using clustering techniques.
• In K Means clustering, the output clusters differ every time we run the algorithm as we begin with a random choice of cluster. On the contrary, in Hierarchical clustering, the results are reproducible in each run.

• K Means clustering requires a pre-hand knowledge of the number of clusters (K) to divide the data into. But, in hierarchical, can stop at whatever the number of clusters we find appropriate with the help of the dendogram.
• The mean shift algorithm often fails at clustering the stray data points, or the ones located between natural clusters.
• In K-Means algorithm, choosing a right cluster number (K) as an input is difficult and end up choosing a sufficiently large cluster number. This will result in situations wherein some natural clusters might be represented by separate multiple clusters. Birch proves to have a better hand when it comes to larger datasets unlike their other counterparts.
• Mean shift performes poorly in these scenarios and the given combinations of features of the dataset mainly due to the inefficient handling of stray data-points.

References

1. D. Cutting, D. Karger, J. Pedersen and J. Tukey, *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, ACM Press, 318– 329 (1992).
2. P. Willett., Information processing & Management, **24**, 577-597, (1988).
3. R. C. Dubes, and A. K. Jain, *Algorithms for Clustering Data*, Prentice Hall (1988).
4. S. Guha, R. Rastogi, and K. Shim, ROCK: *Proceedings of the IEEE International Conference on Data Engineering,* IEEE Explore, Sydney, 512-521 (1999).
5. G. Kowalski, *Information Retrieval Systems – Theory and Implementation*, Kluwer Academic Publishers (1997).
6. B. Larsen, and C. Aone, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD-99, (1999).
7. M. Steinbach, G. Karypis, V. Kumar, *A Comparison of Document Clustering Techniques*, University of Minnesota, Technical Report **34** (2000).
8. Z. Wang, X. Yu, N. Feng, Z. Wang, Journal of Visual Languages and Computing, **25**, 667-675, (2014).
9. R. Katarya, O.P. Verma, Multimedia Tools and Applications, **75**, 9225-9239, (2016).
10. J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Decision Support System, **74**, 12-32, (2015).
11. D. Li, X. Chen, M. Becchi, Z. Zong, *IEEE International Conference on BDClod, SocialCom, SustainCom,* 477-484, (2016).