International Journal of Business Quantitative Economics and Applied Management Research
Volume-4, Issue-12, MAY-2018 ISSN No: 2349-5677

A MACHINE LEARNING ANALYSIS OF MOVIE SIMILARITIES – A CLUSTERING APPROACH

Andy W. Chen
University of British Columbia
Vancouver, Canada

Abstract

In this paper, I build a machine learning model using k-means clustering to group similar movies together. I use the MovieLens 20M dataset which includes more than 10,000 movies, each with relevance scores for more 1,000 tags that describe the characteristics of a movie. The results of the movie are reasonable as by inspection the movies belong to similar genres, themes, and target audience. The results can benefit recommender systems by providing similar movies as recommendations for viewers. The model can be adjusted by changing the number of clusters to group the movies into depending on the need.

IndexTerms*— *Machine Learning, Clustering, Unsupervised Learning,Data Science, Recommendation System

I. INTRODUCTION

The movie industry is booming. Millions of viewers seek information about movies on the Internet. Such service providers rely on recommendation systems to suggest movies a user may also be interested in. Accurate recommendations can enhance the service by saving search time for the users to find movies they may be interested in. Robust recommender systems can also bring more revenues to the website through direct sales of similar products, more traffic coupled with more advertising revenues, and publicity.

In this paper, I use machine learning to explore classification of movies. In particular, I use k-means clustering to cluster movies based on movie characteristics. Movies in the same cluster can be considered similar and be recommended to the users as similar movies they may also be interested in. I find the optimal number of clusters to be 16 using the elbow method. The results are consistent with expectations.

Past research in this area includes the work by Gimet al.[1], who use forecasting methods such as exponential smoothing to predict ratings made by users. Wang et al.[2] makes use of OLAP data warehouses to build a multidimensional model for making movie recommendations. Wei et al.[3] propose a hybrid recommendation system using tags and ratings of movies. Recio-Garcia et al.[4] use a model based on collaborative filtering combined with personality characteristics. Li et al.[5] build a model that incorporates television watching habits and sentiments revealed by users on online blogs to make movie recommendations.

II. METHODS

I use the MovieLens 20M dataset, which includes tags given for 10,993 movies. There are 1,128 tags in total, and each tag has a relevance score given to each of the 10,993 movies. Therefore, there are over 12.4 million relevance scores in total. Each tag is a word used to describe the movie and can be considered a feature of the movie. A tag can be a genre, theme, plot, people, time, or place that appears in the movie. The k-means clustering algorithm will sort the movies into groups based on the relevance scores.

The k-means algorithm is an iterative procedure run over a range of cluster numbers (in my model, I use numbers 1 to 50). For each cluster number, the k-means algorithm starts by assigning a number of movies as cluster centers randomly. Then the algorithm assigns each movie to the closest cluster based on the distance to a cluster center. The new cluster centers are computed by taking the mean values of each feature across all the movies in each cluster. The evaluation metric is the sum of squared differences for all the features between each movie and the cluster center. The above process is repeated until the cluster assignments cannot be improved further. This concludes the clustering for one particular number of clusters.

To complete the entire modeling process, the above process is repeated over a range of cluster values. The sum of squared errors is plotted over the range of cluster numbers. As the number of clusters increases, the sum of squared errors decreases. However, the decrease in error may be not significant enough for the additional cluster to be added. In other words, there is a point where any additional cluster added does not improve the errors by a worthwhile amount. The optimal number of clusters is chosen using the elbow method. The idea is to choose the number of clusters where the last large reduction in errors occurred. This particular number of clusters should be evident in a plot of the errors against number of clusters.

III. RESULTS AND DISCUSSION

Figure 1 shows the plot of sum of squared errors against number of clusters. Table 1 shows the actual values of the sum of squared errors as number of clusters increases. Using the elbow method, I choose the optimal number of clusters to be 16, which is the highest number that gives a reduction of more than 1000 in the errors. The reduction in errors from 19 to 20 is also greater than 1000, but the reductions before 20 clusters are not, so I choose 16 instead of 20. Table 2 shows the number of movies in each of the 16 clusters.

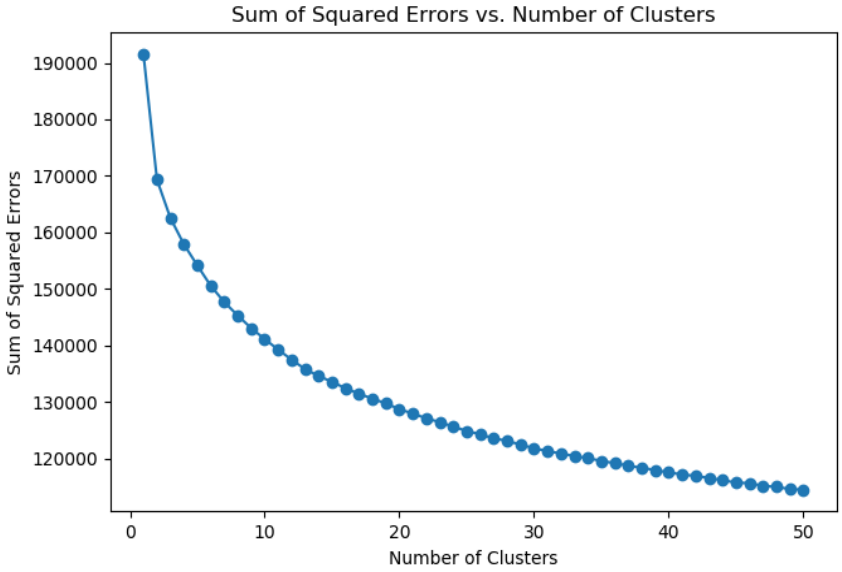

Fig. 1. Sum of Squared Errors vs. Number of Clusters

TABLE I. SUM OF SQUARED ERRORS VS. NUMBER OF CLUSTERS

Number of Clusters	Sum of Squared Errors	Decrease in SSE
1	191477	-
2	169312	-22165
3	162522	-6789
4	157877	-4646
5	154131	-3746
6	150511	-3620
7	147653	-2858
8	145275	-2378
9	143034	-2241
10	141106	-1927
11	139240	-1866
12	137460	-1780
13	135785	-1675
14	134581	-1204
15	133536	-1045
16	132401	-1135
17	131441	-959
18	130548	-893
19	129733	-815
20	128701	-1031
21	127919	-783
22	127090	-828
23	126399	-691

24	125622	-777
25	124842	-780
26	124269	-574
27	123540	-729
28	123091	-449
29	122422	-669
30	121789	-633
31	121274	-515
32	120859	-414
33	120398	-461
34	120042	-356
35	119537	-505
36	119156	-381
37	118668	-488
38	118336	-333
39	117910	-426
40	117522	-388
41	117123	-399
42	116910	-212
43	116501	-410
44	116154	-346
45	115691	-464
46	115620	-71
47	115136	-484
48	114950	-186
49	114565	-385
50	114446	-120

TABLE II. NUMBER OF MOVIES IN EACH CLUSTER

Cluster	Number of Movies
1	775
2	586
3	872
4	657
5	604
6	518
7	1529
8	502
9	703
10	661
11	394
12	461
13	500
14	1409

15	238
16	584

The results of the clustering model are reasonable. Table 3 shows a subset of the movies from 3 clusters. In cluster 1, most movies belong to the adventure and thriller categories and include themes such as death and violence. In cluster 2, most movies have the theme of supernatural beings and belong to the fantasy genre. Most movies in cluster 3 are animated movies for children and family. Depending on the needs of the recommendation system, one may decide to group the movies into more clusters. In that case, the classification may be more suitable, but viewers will be given a smaller set of recommendations.

TABLE III. SAMPLE OF MOVIES IN CLUSTERS

Cluster 1	Cluster 2	Cluster 3
Sudden Death (1995)	X-Men (2000)	Toy Story (1995)
Golden Eye (1995)	E.T. the Extra-Terrestrial (1982)	Jumanji (1995)
Cutthroat Island (1995)	Jurassic Park (1993)	Balto (1995)
Die Hard: With a Vengeance (1995)	Alien (1979)	Babe (1995)
Assassins (1995)	Terminator 2: Judgment Day (1991)	Pocahontas (1995)

IV. CONCLUSION

This paper explores the use of clustering using the k-means algorithm in grouping movies. The results can be useful for recommendation systems in a variety of services that offer movie information for potential movie viewers. The model is adaptable in the sense that the number of clusters can be adjusted to suit particular needs. A future extension could be to include more variety of movie characteristics such as cast members, runtime, and director.

REFERENCES

[1] Gim G, Jeong H, Lee H, Yun D. Group Recommendation in Context. CAMRa '11 Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation. 2011:11-14.

[2] Wang SS, Lin B, Chen WT. Using Contextual Information and Multidimensional Approach for Recommendation. Expert Systems with Applications. 2009;36(2):1268-1279.

[3] Wei S, Zheng X, Chen D, Chen C. A Hybrid Approach for Movie Recommendation Via Tags and Ratings. Electronic Commerce Research and Applications. 2016;18:83-94.

[4] Recio-Garcia JA, Jimenez-Dias G, Sanchez-Ruiz AA, Diaz-Agudo B. Personality Aware Recommendations to Groups. RecSys '09 Proceedings of the third ACM conference on Recommender systems. 2009:325-328.

[5] Li H, Cui J, Shen B, Ma J. An Intelligent Movie Recommendation System Through Group-Level Sentiment Analysis in Microblogs. Neurocomputing. 2016;210:164-173.