Application of von Mises-Fisher mixture model for text clustering IMDB movie synopses

Justin Peter

April 11, 2022

Abstract

Finite mixture model is an effective tool for modeling complex, heterogeneous data. Clustering on unit hypersphere using finite mixture of von Mises-Fisher Distributions was introduced by (1). This model is known to be effective for clustering text data. In this paper, we will describe the application of von Mises-Fisher mixture model on movie description of crime genre from the internet movies database (IMDB). The objective is to identify movies of similar types within the crime genre and observe the change in movie content over time.

Keywords: text mining, mixture models, clustering, vMF distributions

1 Introduction

Text mining, otherwise known as text analytics, has the ability to transform text into structured data that can be suitable for data analysis algorithms, like clustering. We have extracted information from one of the internet movies database (IMDB) where people submit their ratings and comments to movies they may have watched. Each movie contains a description. We focused on crime genre and extracted the description of the movies with user rating greater than or equal to 8. The goal is to find cluster of movies within the crime genre.

2 Methodology

2.1 Clustering

The objective of cluster analysis is to form distinct group of data points similar in feature. These groups are called clusters. The data points that belong to a particular cluster are more similarly compared to a group of data points that would belong to a different cluster. There are many different algorithms when it comes to clustering and some examples include *hierarchical*, *k-means*, and *model-based* clustering.

A finite mixture model (2) is a convex combination of several probability distributions known as mixture components. Suppose $\underline{\boldsymbol{x}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ be an observed sample of n independent realization from the probability density function given by

$$g(\boldsymbol{x}; \boldsymbol{\Psi}) = \sum_{k=1}^{K} \alpha_k f_k(\boldsymbol{x}; \boldsymbol{\Psi}_k). \tag{1}$$

Equation 1 is known as a finite mixture model with parameter vector $\boldsymbol{\Psi}$. Here, K represents the number of components, also known as mixture order. $f_k(\boldsymbol{x}; \boldsymbol{\Psi}_k)$ is the pdf of the k^{th} mixture component with parameter $\boldsymbol{\Psi}_k$ and α_k is the corresponding mixing proportion with restriction $\sum_{k=1}^{K} \alpha_k = 1$. The estimation of $\boldsymbol{\Psi}^\top = \{(\alpha_k, \boldsymbol{\Psi}_k^\top), k = 1, 2, \ldots, K\}$ is usually done by employing the expectation-maximization (EM) algorithm (3) which is an iterative method of finding maximum likelihood estimates (MLE).

Model based clustering assumes a one to one correspondence between mixture components and clusters. Each mixture component is considered to be responsible for modeling a cluster.

2.2 vMF Distribution

For clustering text data, the use of von Mises Fisher (vMF) mixture model is popular in literature. The pdf of vMF distribution is given by,

$$f(x|\theta) = e^{\theta^T x} / {}_0F_1(; d/2; \|\theta\|^2 / 4) \tag{2}$$

where

$${}_0F_1(; v; z) = \sum_{n=0}^{\infty} \frac{\Gamma(v)}{\Gamma(v+n)} \frac{z^n}{n!}$$

$$ {}_0F_1(;v+1;z^2/4) = \frac{I_v(z)\Gamma(v+1)}{(z/2)^v} $$

.

An alternative parameterization of vMF distribution is often convenient and is given by $\theta = \kappa\mu$, where $\kappa = \|\theta\|$ is the concentration parameter and μ is the mean direction parameter. This yields,

$$ C_d(\kappa) = 1/{}_0F_1(;d/2;\kappa^2/4) $$

$$ f(x|\theta) = C_d(\|\theta\|)e^{\theta^T x} $$

.

In this study, we consider a finite mixture model of vMF distribution for finding clusters in text data. Such model is obtained by replacing the mixture component in 1 by the density defined in 2.

2.3 Data Manipulation

The dataset was extracted from the following link, CRIME. The dataset contains at least 30000 movies and 10 columns, including movie name, year produced, type of genre, description, etc. From here, we filtered out the data to all movies that contain crime, because for each movie, there are at least 2 different genres that describe it. After we filter out the data to crime, we filter out all the movies that have at least a rating of 8. Then, we come to find out that the number of movies decrease from over 30,000 to around 400.

2.4 MovMF

We start by transforming the description into a document term matrix, which describes the frequency of every unique word in each movie description. We computed weighted term frequency and inverse document frequency which shows the importance of a word in the movie descriptions. This calculates the occurrences a word appears in one movie description over all the movie descriptions that contain that word. The document term matrix also includes tokenization, stop words, stem words, and word lengths. Stop words include words that we consider filler words, such as *and*, *the*, *for*, *to*, etc. Stop words depend on the type of data and objective. For our data, we remove stop words because the redundancy of filler words added to the algorithm renders it difficult to identify cluster categories. Stemming reduces words to their

root forms to allow for the normalization of these words. When applied to data, stemming removes vowels rather than consonants, so generally it doesn't help the algorithm because the inflection of words will be included in the clusters instead of the actual words which render it difficult to identify cluster categories. Word lengths refer to the length of the character string of each word, so the characters are limited to at least 3 for each word.

When we created the document term matrix, we observed that there are words that aren't stop words, but are useless. For example, film appeared a lot, but because film refers to movies, the word doesn't improve the algorithm. We performed various word selections to decide word removal. The first is a frequency word removal by taking all descriptions into a data frame, activate every unique word, remove stop words, and order words by frequency. After we remove words that have the lowest and highest frequencies, there's a slight improvement on the model so we use a different word removal approach.

The second approach is a manual removal of words by going through each word and decide whether to include it. If we observe combinations of words like *blood* and *bloody* or *gang*, *gangs*, and *gangsters*, we treat them as the same word. *Bloody* would become *blood* and *gangs* along with *gangsters* would be gang. If we observe similar words in context, we keep all of them. For example, if we check the word *father*, which indicates a member of a house or family, we keep similar words like *mother*, *son*, and *daughter*.

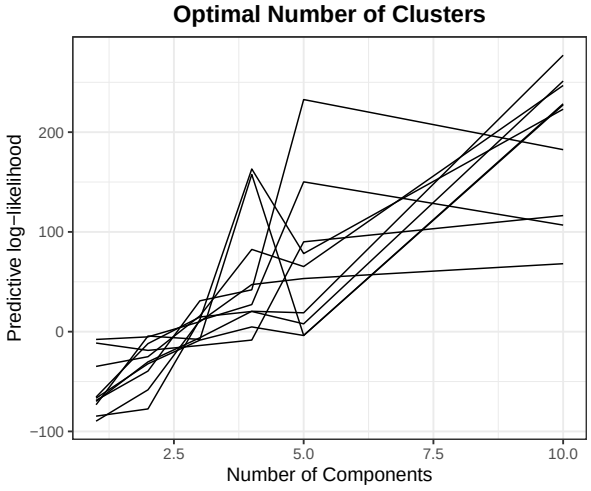

Figure 1: Plot of number of clusters vs predictive log likelihood of 10 folds

Figure 1 represents a plot of the number of clusters vs the predictive log likelihood of 10 cross validation folds. We use the movMF package (4) for text clustering which uses finite mixtures of vMF distribution. After processing the data performed clustering and since the number of clusters is unknown, we used a cross validation method. The plot was formed by applying the mixture model distribution on 10 folds, and then computing the predictive log likelihood of every mixture component for each of the 10 folds. The predictive log likelihood is expected to be the highest for the correct mixture order. We observe that 7 out of 10 folds have the highest predictive log likelihood when number of components is equal to 4. Hence, 4 is considered to be the optimal mixture order.

3 Results

Cluster 1	Cluster 2	Cluster 3	Cluster 4
murder	police	man	crime
people	story	young	life
city	family	woman	friend
love	brother	revenge	boy
son	team	life	old
journey	father	girl	mother
journalist	cop	private	school
accused	officer	house	family
blood	gang	lawyer	boss
children	killer	husband	syndicate

Table 1: Application of movMF algorithm to create clusters

3.1 Creation of Clusters

The table 1 was formed by taking the movMF algorithm and using the optimal mixture order 4. To decide on cluster names: we put like words together and do bigram frequencies to find support words. To put like words together, we search for words within the group that look like a connection. For example, accused and journalist: a journalist could be investigating a crime or

could be involved in one. To do a bigram frequency, we find frequency of pair of words in decreasing order and observe which words associate with those from our group. For example, murder appears with trial in high frequency, so this could imply a trial based on a murder case.

3.2 Cluster Names

After inspecting the keywords with each cluster, we propose the following names for each cluster. Cluster 1 is **Investigative Journalism/City Crimes**, which is about crime investigations. Like words include journalist and accused, blood and children, people and city. Bigrams imply that murder connects with investigator, blood connects with family, journey connects with bus. Cluster 2 is **Police Procedural/Family Crimes**, which involves police and unorganized gang related activities. Like words include police and officer, gang and killer, family and team. Bigrams imply brother connects with mafia, family and team connect with assassin, gang connects with violent, killer connects with serial. Cluster 3 is **Relationships/Courtroom**, which is about family related issues involving a lawyer or investigator. Like words include man and woman, young and house, private and lawyer. Bigrams imply that man connects with innocent, woman connects with daughter, revenge connects to gang. Cluster 4 is **Organized Crimes/Coming of Age**, which is about family run business centered on crime and coming of age stories. Like words include boy and school, crime and syndicate, boss and family. Bigrams imply that school connects with brother, boss connects to loyalty, and boy, boss, family, friend, and syndicate all relate to crime.

3.3 Cluster Graphs

In Figure 2 we observe the most frequent secondary genres are drama, action, thriller, mystery, and comedy. In Figure 3, the highest range of frequencies in secondary genres take place between 1990-2020. In Figure 4, all clusters have the highest frequencies between 2010-2020. Police Procedural/Family Crimes has the highest combination of frequencies from 1990-2020, just like Figure 3. Figure 5 display that Investigative Journalism/City Crimes connect with murder. Organized Crimes/Coming of Age connect with crime, friend, and life. Police Procedural/Family Crimes connect with gang, family, police, and story. Relationships/Courtroom connect with young and man.

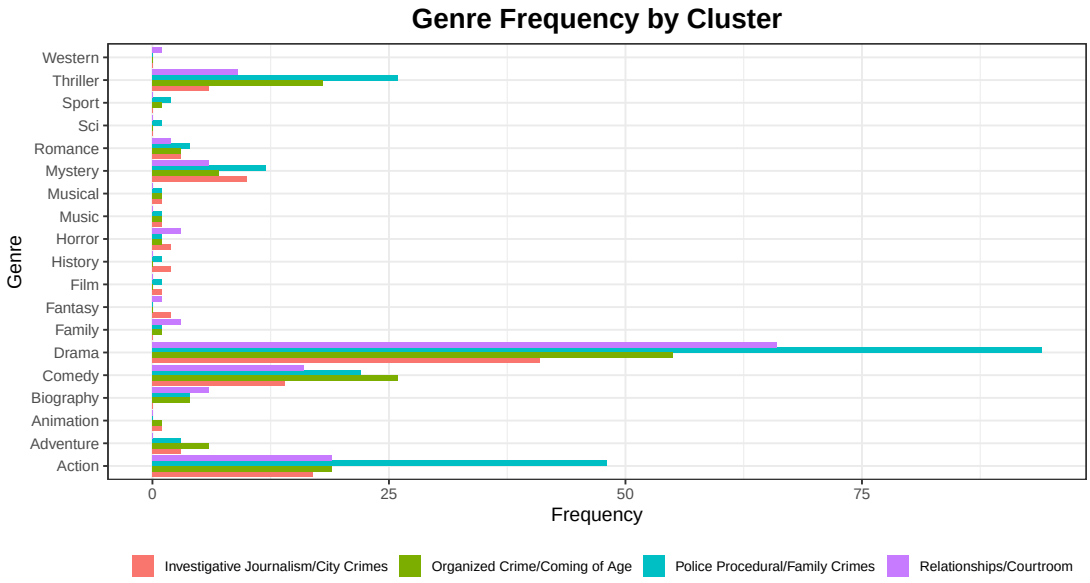

Figure 2: Graph shows the frequency of genres apart from crime by cluster

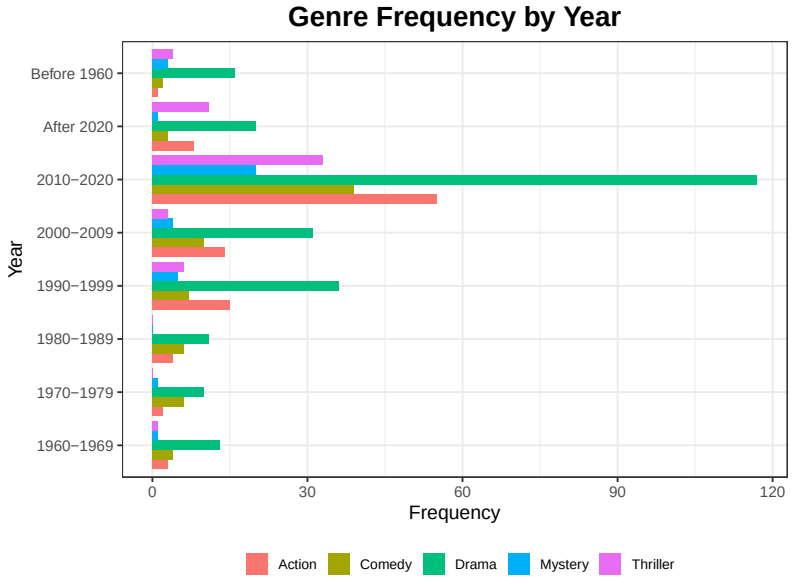

Figure 3: Graph shows the frequency of the top 5 sub-genres by each decade

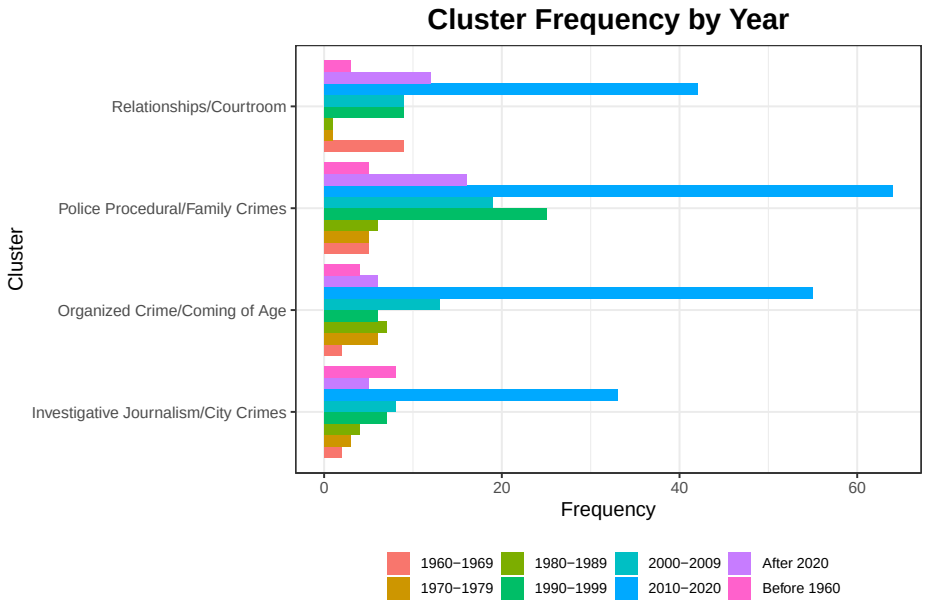

Figure 4: Graph shows the frequency of every cluster by each decade

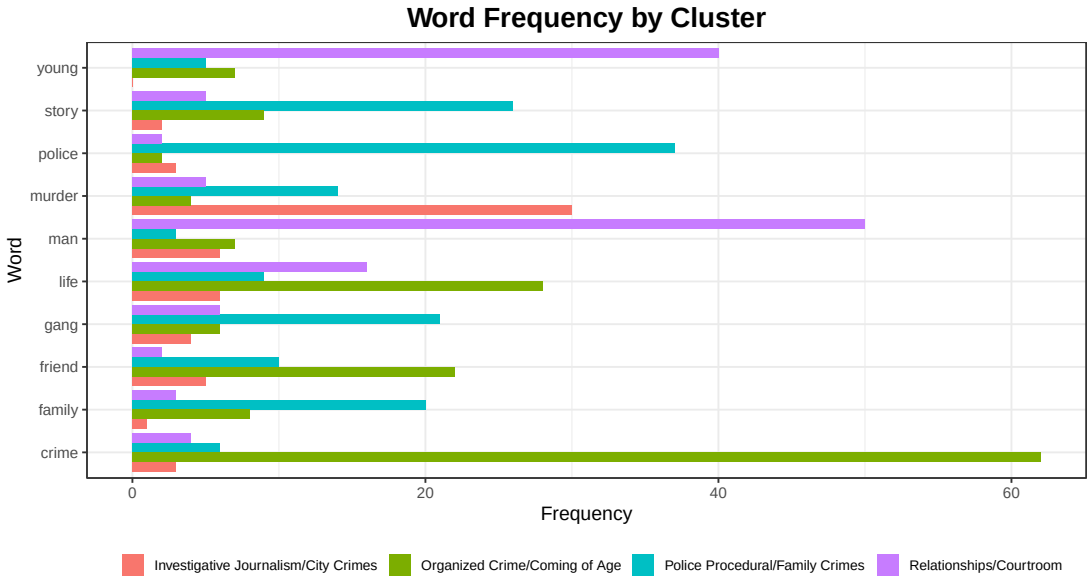

Figure 5: Graph shows the frequency of the top 10 words for each cluster

Movie	Year	Sub-Genres	Cluster
The Dark Knight	2008	Action, Drama	Investigative Journalism/City Crimes
The Dark Knight Rises	2012	Action	Investigative Journalism/City Crimes
The Green Mile	1999	Drama, Fantasy	Investigative Journalism/City Crimes
Taxi Driver	1976	Drama	Investigative Journalism/City Crimes
12 Angry Men	1957	Drama	Investigative Journalism/City Crimes
Pulp Fiction	1994	Drama	Police Procedural/Family Crimes
The Godfather	1972	Drama	Police Procedural/Family Crimes
Se7en	1995	Drama, Mystery	Police Procedural/Family Crimes
The Silence of the Lambs	1991	Drama, Thriller	Police Procedural/Family Crimes
The Departed	2006	Action, Drama	Police Procedural/Family Crimes
Catch Me If You Can	2002	Biography, Drama	Relationships/Courtroom
To Kill a Mockingbird	1962	Drama	Relationships/Courtroom
Cool Hand Luke	1967	Drama	Relationships/Courtroom
In the Name of the Father	1993	Biography, Drama	Relationships/Courtroom
Memories of Murder	2003	Drama, Mystery	Relationships/Courtroom
The Wolf of Wall Street	2013	Biography, Comedy	Organized Crimes/Coming of Age
The Godfather: Part II	1974	Drama	Organized Crimes/Coming of Age
Joker	2019	Drama, Thriller	Organized Crimes/Coming of Age
Goodfellas	1990	Biography, Drama	Organized Crimes/Coming of Age
The Grand Budapest Hotel	2014	Adventure, Comedy	Organized Crimes/Coming of Age

Table 2: Top 5 Popular Movies by Cluster

3.4 Table of Movies

The table 2 was formed by obtaining the top 5 movies with the most votes for each cluster. For Investigative Journalism/City Crimes, each movie was produced in a different decade. The two most common genres from those movies are action and drama. The Dark Knight and The Dark Knight Rises are examples of city crimes because the main idea revolves around Batman saving the people of Gotham City from Joker and Bane, so key words include murder, people, and city. For Police Procedural/Family Crimes, most movies take place in the 90s and drama is the most common genre. The Godfather is an example of family crimes because the plot is based on the patriarch of the Corleone family focusing on transforming his son to a mafia boss, so key words include family and son. For Relationships/Courtroom, a majority of these movies take place in the decades 1960-1970 and 2000-2010. The two most common genres are biography and drama. In the Name of the Father is an example of Courtroom because the story is on a forced confession of a man to a bombing and the man hires a lawyer to fight the battle and save him and his imprisoned father, so key words are man, father, and lawyer.

Organized Crimes/Coming of Age contains most movies between 2010-2020 with common genres like biography, comedy, and drama. The Godfather Part 2 is an example of both organized crimes and coming of age because the movie simultaneously focuses on the events of the son building up his crime business syndicate and covers the journey of the patriarch from childhood to building the family business, thereby creating a major difference to its previous installment with key words family, crime, syndicate, and life.

4 Conclusion

4.1 Potential

Should we rely solely on word selection to improve our algorithm? This can be complex when there are unique words and combinations of the same word. It's tricky to analyze connections between words if the data contains many observations. Another risk would be accepting words without understanding the context in which the word is used. For example, Memories of Murder from table 2 is about two detectives investigating a case of rape on multiple young women. The context implies that the movie should be placed in Investigative Journalism, but due to the frequency of the words young and women, the movie was placed on Relationships/Courtroom. There could be details within that movie that make it Relationships/Courtroom, but the problem is when we select words, should we analyze each one by context? It would be very time consuming to analyze each word in context and check every movie description. The algorithm responded well for the majority of movies, however it would be helpful if there was a function that can detect the weights of unique words in different movie descriptions in order to build up the selection process.

4.2 R Libraries

We utilized many libraries in R. Dplyr allows for manipulation of data. Rvest allows for web scraping. XML2 allows for parsing of HTML. StringR allows for manipulation of character strings. TM allows for text mining techniques, like removal of words and formation of document term matrix. SnowballC allows for word stemming. MovMF allows for clustering of text data.

References

[1] A.Banerjee, I.S.Dhillon, J.Ghosh, S.Sra, and G.Ridgeway,"Clustering on the unit hypersphere using von mises-fisher distributions." Journal of Machine Learning Research, vol. 6, no. 9, 2005.

[2] S. Sarkar, V. Melnykov, and R. Zheng, "Gaussian mixture modeling and model-based clustering under measurement inconsistency," Advances in Data Analysis and Classification, vol. 14, no. 2, pp. 379–413, 2020.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood for incomplete data via the EM algorithm (with discussion)," Journal of the Royal Statistical Society, Series B, vol. 39, pp. 1–38, 1977.

[4] K. Hornik and B. Grün, "movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions," Journal of Statistical Software, vol. 58, no. 10, 2014.