



DIMENSIONLESS
TECHNOLOGY

Clustering

Recommendations Worth a Million

An Introduction to Clustering

Netflix

- Online DVD rental and streaming video service
- More than 40 million subscribers worldwide
- \$3.6 billion in revenue
- Key aspect is being able to offer customers accurate movie recommendations based on a customer's own preferences and viewing history



The Netflix Prize

- From 2006 – 2009 Netflix ran a contest asking the public to submit algorithms to predict user ratings for movies
- Training data set of $\sim 100,000,000$ ratings and test data set of $\sim 3,000,000$ ratings were provided
- Offered a grand prize of \$1,000,000 USD to the team who could beat Netflix's own algorithm, Cinematch, by more than 10%, measured in RMSE

Contest Rules



- If the grand prize was not yet reached, progress prizes of \$50,000 USD per year would be awarded for the best result so far, as long as it had $>1\%$ improvement over the previous year.
- Teams must submit code and a description of the algorithm to be awarded any prizes
- If any team met the 10% improvement goal, last call would be issued and 30 days would remain for all teams to submit their best algorithm.

Initial Results



- The contest went live on October 2, 2006
- By October 8, a team submitted an algorithm that beat Cinematch
- By October 15, there were three teams with algorithms beating Cinematch
- One of these solutions beat Cinematch by $>1\%$, qualifying for a progress prize

Progress During the Contest

- By June 2007, over 20,000 teams had registered from over 150 countries
- The 2007 progress prize went to team BellKor, with an 8.43% improvement on Cinematch
- In the following year, several teams from across the world joined forces

Competition Intensifies

- The 2008 progress prize went to team BellKor which contained researchers from the original BellKor team as well as the team BigChaos
- This was the last progress prize because another 1% improvement would reach the grand prize goal of 10%

Last Call Announced

- On June 26, 2009, the team BellKor's Pragmatic Chaos submitted a 10.05% improvement over Cinematch

Netflix Prize

[Home](#)
[Rules](#)
[Leaderboard](#)
[Register](#)
[Update](#)
[Submit](#)
[Download](#)

Leaderboard

10.05%

Display top

 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52

Last 30 days



- Other teams had 30 days to submit algorithms before the contest closed.
- These 30 days were filled with intense competition and even more progress.
- But before revealing what happened, let's investigate how we could try to predict user ratings.

We'll discuss how recommendation systems work

Quick Question

- About how many years did it take for a team to submit a 10% improvement over Cinematch?

☐ 0.5

☐ 1.5

☐ 2.5

☐ 3.5

Recommendation Systems

Predicting the Best User Ratings




- Netflix was willing to pay over \$1M for the best user rating algorithm, which shows how critical the recommendation system was to their business
- What data could be used to predict user ratings?
- Every movie in Netflix's database has the ranking from all users who have ranked that movie
- We also know facts about the movie itself: actors, director, genre classifications, year released, etc.

Using Other Users' Rankings

	Men in Black	Apollo 13	Top Gun	Terminator
Amy	5	4	5	4
Bob	3		2	5
Carl		5	4	4
Dan	4	2		

- Consider suggesting to Carl that he watch “Men in Black”, since Amy rated it highly and Carl and Amy seem to have similar preferences
- This technique is called **Collaborative Filtering**

Using Movie Information

- We saw that Amy liked “Men In Black”
 - It was directed by Barry Sonnenfeld 
 - Classified in the genres of action, adventure, sci-fi and comedy 
 - It stars actor Will Smith 
- Consider recommending to Amy:
 - Barry Sonnenfeld’s movie “Get Shorty”
 - “Jurassic Park”, which is in the genres of action, adventure, and sci-fi
 - Will Smith’s movie “Hitch”

This technique is called **Content Filtering**

Strengths and Weaknesses

- Collaborative Filtering Systems
 - Can accurately suggest complex items without understanding the nature of the items
 - Requires a lot of data about the user to make accurate recommendations
 - Millions of items – need lots of computing power
- Content Filtering
 - Requires very little data to get started
 - Can be limited in scope

Hybrid Recommendation Systems

- Netflix uses both collaborative and content filtering
- For example, consider a collaborative filtering approach where we determine that Amy and Carl have similar preferences.
- We could then do content filtering, where we would find that “Terminator”, which both Amy and Carl liked, is classified in almost the same set of genres as “Starship Troopers”
- Recommend “Starship Troopers” to both Amy and Carl, even though neither of them have seen it before

Quick Question

Let's consider a recommendation system on Amazon.com, an online retail site.

If Amazon.com constructs a recommendation system for books, and would like to use the same exact algorithm for shoes, what type would it have to be?

☐ Collaborative Filtering

☐ Content Filtering

Quick Question

- If Amazon.com would like to suggest books to users based on the previous books they have purchased, what type of recommendation system would it be?

☐ Collaborative Filtering

☐ Content Filtering

Clustering

MovieLens Data



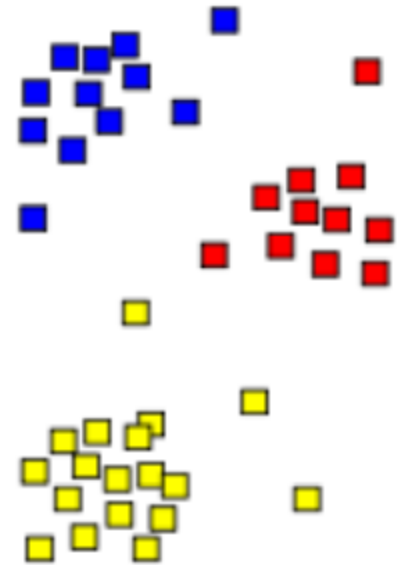
- www.movielens.org is a movie recommendation website run by the GroupLens Research Lab at the University of Minnesota
- They collect user preferences about movies and do collaborative filtering to make recommendations
- We will use their movie database to do content filtering using a technique called clustering

MovieLens Item Dataset

- Movies in the dataset are categorized as belonging to different genres
 - (Unknown) • Action • Adventure • Animation • Children's
 - Comedy • Crime • Documentary • Drama • Fantasy
 - Film Noir • Horror • Musical • Mystery • Romance
 - Sci-Fi • Thriller • War • Western
- Each movie may belong to many genres
- Can we systematically find groups of movies with similar sets of genres?

Why Clustering?

- “Unsupervised” learning
 - Goal is to segment the data into similar groups instead of prediction
- Can also cluster data into “similar” groups and then build a predictive model for each group
 - Be careful not to overfit your model!
This works best with large datasets



Types of Clustering Methods

- There are many different algorithms for clustering
 - Differ in what makes a cluster and how to find them
- We will cover
 - Hierarchical
 - K-means in the next lecture

Quick Question



In the previous slides, we discussed how clustering is used to split the data into similar groups. Which of the following tasks do you think are appropriate for clustering? Select all that apply.

- ☐ Dividing search results on Google into categories based on the topic
- ☐ Grouping players into different "types" of basketball players that make it to the NBA
- ☐ Predicting the winner of the Major League Baseball World Series

Computing Distances

Distance Between Points

- Need to define distance between two data points
 - Most popular is “Euclidean distance”
 - Distance between points i and j is

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ik} - x_{jk})^2}$$

where k is the number of independent variables

Distance Example

- The movie “Toy Story” is categorized as Animation, Comedy, and Children’s
 - Toy Story:
(0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0)
- The movie “Batman Forever” is categorized as Action, Adventure, Comedy, and Crime
 - Batman Forever:
(0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0)



Distance Between Points

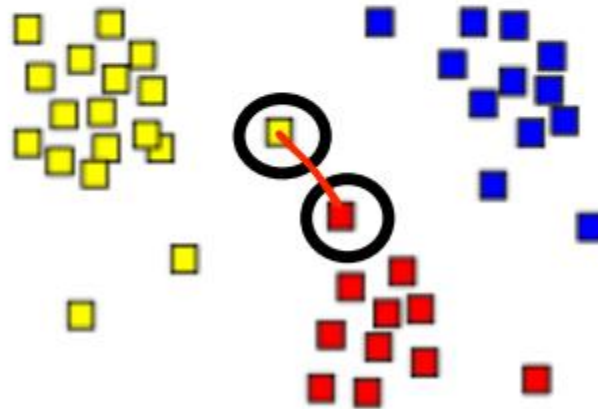
- Toy Story: (0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0)
- Batman Forever: (0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0)

$$d = \sqrt{(0-0)^2 + (0-1)^2 + (0-1)^2 + (1-0)^2 + \dots}$$
$$= \sqrt{5}$$

- Other popular distance metrics:
 - Manhattan Distance
 - Sum of absolute values instead of squares
 - Maximum Coordinate Distance
 - Only consider measurement for which data points deviate the most

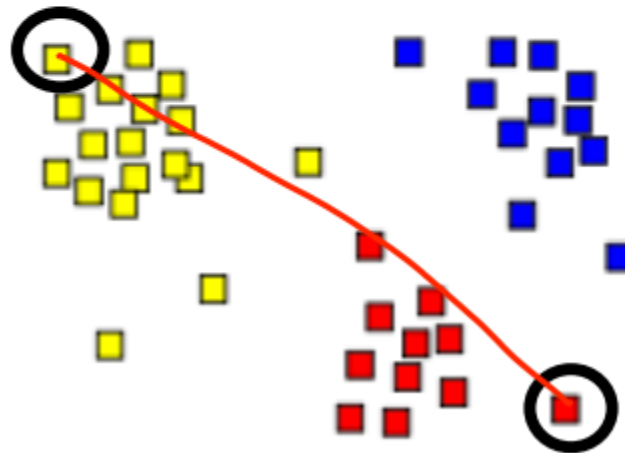
Distance Between Clusters

- Minimum Distance
 - Distance between clusters is the distance between points that are the closest



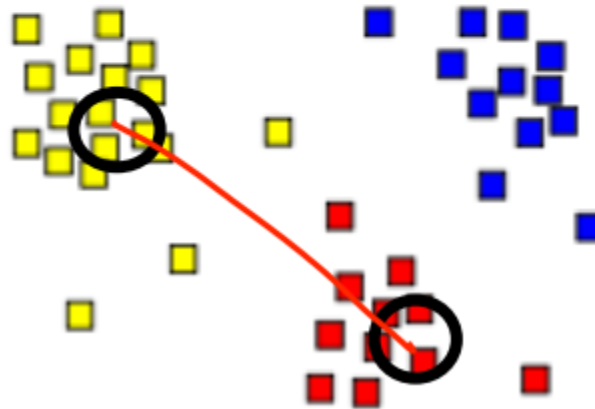
Distance Between Clusters

- Maximum Distance
 - Distance between clusters is the distance between points that are the farthest



Distance Between Clusters

- Centroid Distance
 - Distance between centroids of clusters
 - Centroid is point that has the average of all data points in each component



Normalize Data



- Distance is highly influenced by scale of variables, so customary to normalize first
- In our movie dataset, all genre variables are on the same scale and so normalization is not necessary
- However, if we included a variable such as “Box Office Revenue,” we would need to normalize.

Quick Question

The movie "The Godfather" is in the genres action, crime, and drama, and is defined by the vector:

$(0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

The movie "Titanic" is in the genres action, drama, and romance, and is defined by the vector:

$(0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

What is the distance between "The Godfather" and "Titanic", using euclidean distance?

Hierarchical Clustering

Hierarchical

- Start with each data point in its own cluster



Hierarchical



- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical

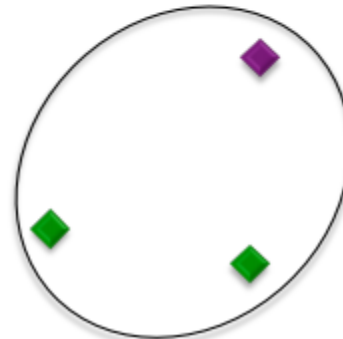
- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical



- Combine two nearest clusters (Euclidean, Centroid)



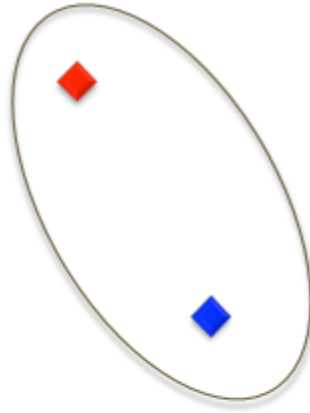
Hierarchical

- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical

- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical

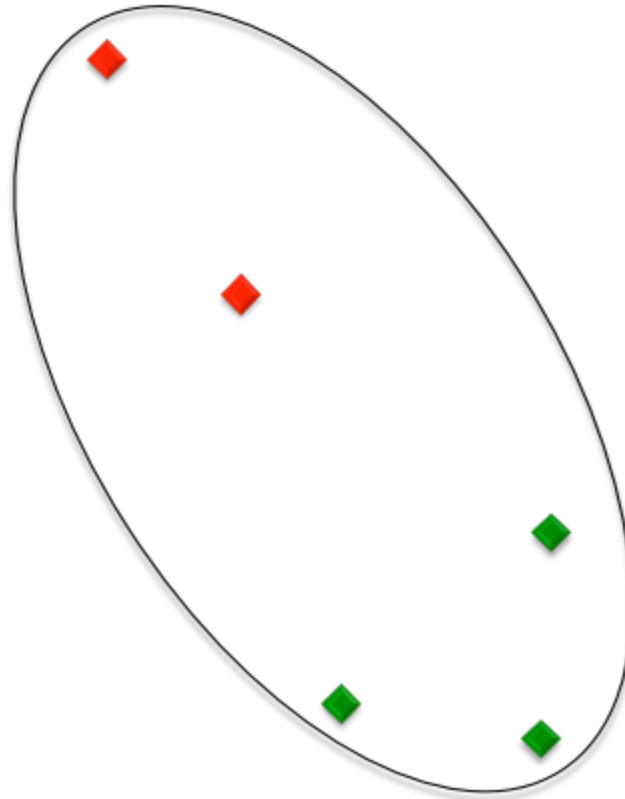
- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical



- Combine two nearest clusters (Euclidean, Centroid)



Hierarchical

- Combine two nearest clusters (Euclidean, Centroid)

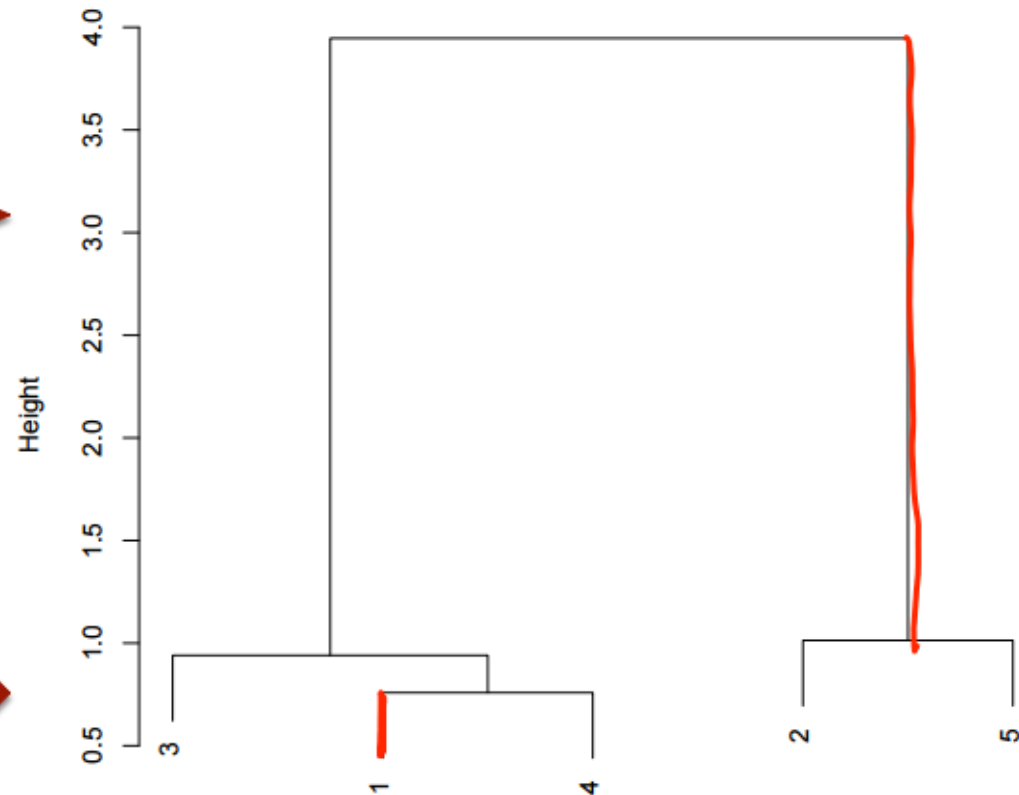


Display Cluster Process

Height of vertical lines represents distance between points or clusters

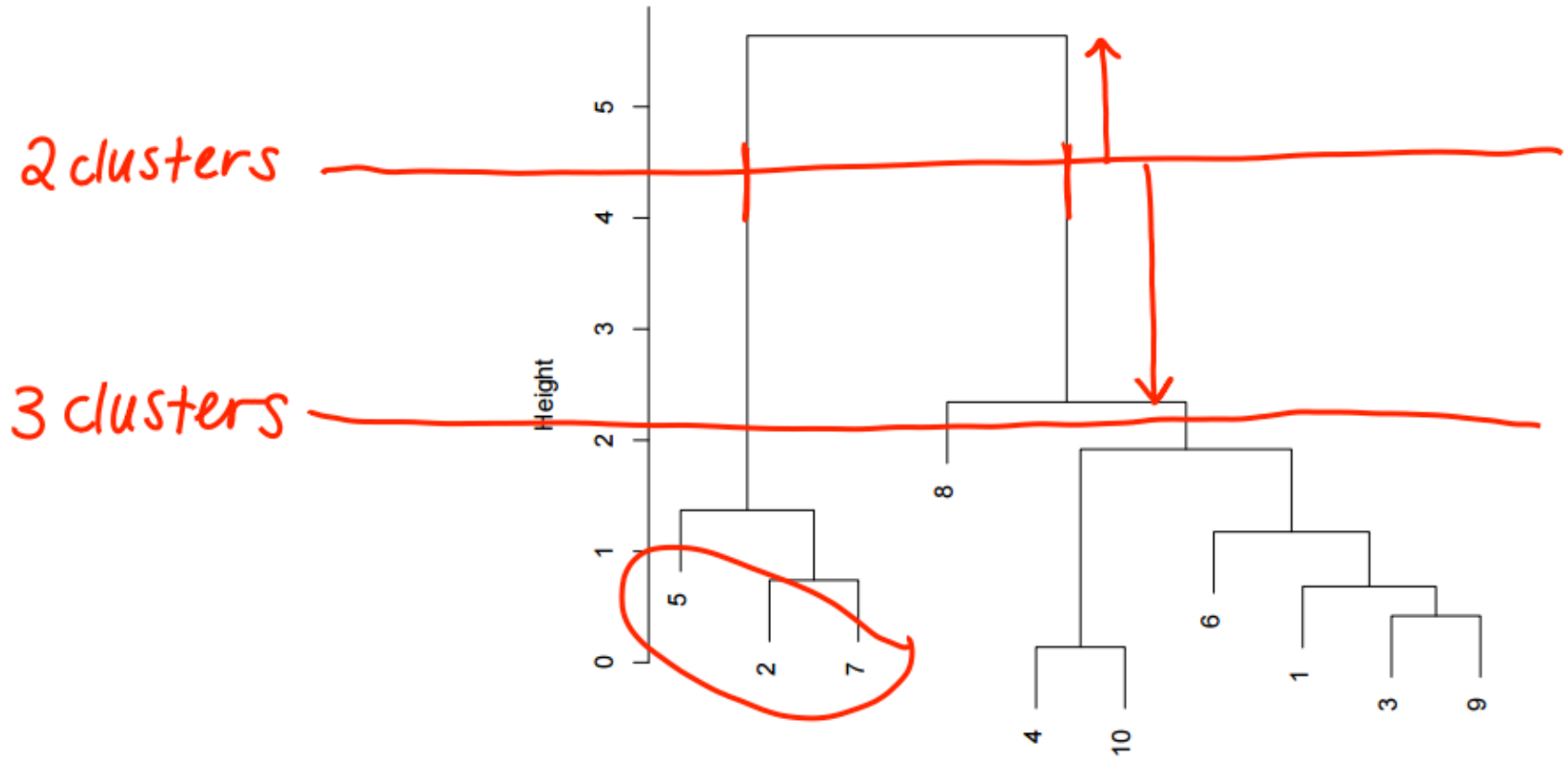
Data points listed along bottom

Cluster Dendrogram



Select Clusters

Cluster Dendrogram



Meaningful Clusters?

- Look at statistics (mean, min, max, . . .) for each cluster and each variable
- See if the clusters have a feature in common that was not used in the clustering (like an outcome)

Quick Question



Suppose you are running the Hierarchical clustering algorithm with 212 observations.

How many clusters will there be at the start of the algorithm?

How many clusters will there be at the end of the algorithm?

Getting the Data

Data



- Open the link
<http://files.grouplens.org/datasets/movie-lens/ml-100k/u.item>
- Copy the complete text and paste it in any word editor(Note pad).
- Our data is not a CSV file. It's a text file, where the entries are separated by a vertical bar.
- Import this file in R using `read.table()`
- We'll call our dataset "movies".
 - `movies<-read.table ("u.item.txt", header = FALSE, sep = "|", quote="")`

Structure



- Take a look at the structure of our data using the `str` function.
- We have 1,682 observations of 24 different variables.
- There are no names of the columns, so we will give them
 - `colnames(movies)=c("ID", "Title", "ReleaseDate", "VideoReleaseDate", "IMDB", "Unknown", "Action", "Adventure", "Animation", "Children", "Comedy", "Crime", "Documentary", "Drama", "Fantasy", "FilmNoir", "Horror", "Musical", "Mystery", "Romance", "SciFi", "Thriller", "War", "Western")`

Removing the not required

- Now look at the dataset again.
- We have the same number of observations and the same number of variables, but each of them now has the name that we just gave.
- We won't be using the ID, release date, video release date, or IMDB variables.
- Remove them from the dataset "movies".
 - `movies$ID=NULL`
 - `movies$ReleaseDate=NULL`
 - `movies$VideoReleaseDate=NULL`
 - `movies$IMDB=NULL`

Removing the duplication

- There are a few duplicate entries in our data set.
- We need to remove them.
 - `movies<-unique(movies)`
- Look at the structure
- Now, we have 1,664 observations, a few less than before, and 20 variables.

Quick Question

Using the table function in R, please answer the following questions about the dataset "movies".

How many movies are classified as comedies?



How many movies are classified as westerns?



How many movies are classified as romance AND drama?



Hierarchical Clustering in R

Clustering



- In this lecture we'll use hierarchical clustering to cluster the movies in the Movie Lens data set by genre
- After we make our clusters, we'll see how they can be used to make recommendations
- There are two steps to hierarchical clustering
 - First we have to compute the distances between all data points
 - Second, we need to cluster the points

Distance



- To compute the distances we can use the `dist` function.
- We only want to cluster our movies on the genre variable, not on the title variable, so we'll calculate the distance on columns two through 20.
- The second argument is `method="euclidean"`, meaning that we want to use euclidean distance.
 - `Distance<-dist(movies[[2:20]],method = "euclidean")`

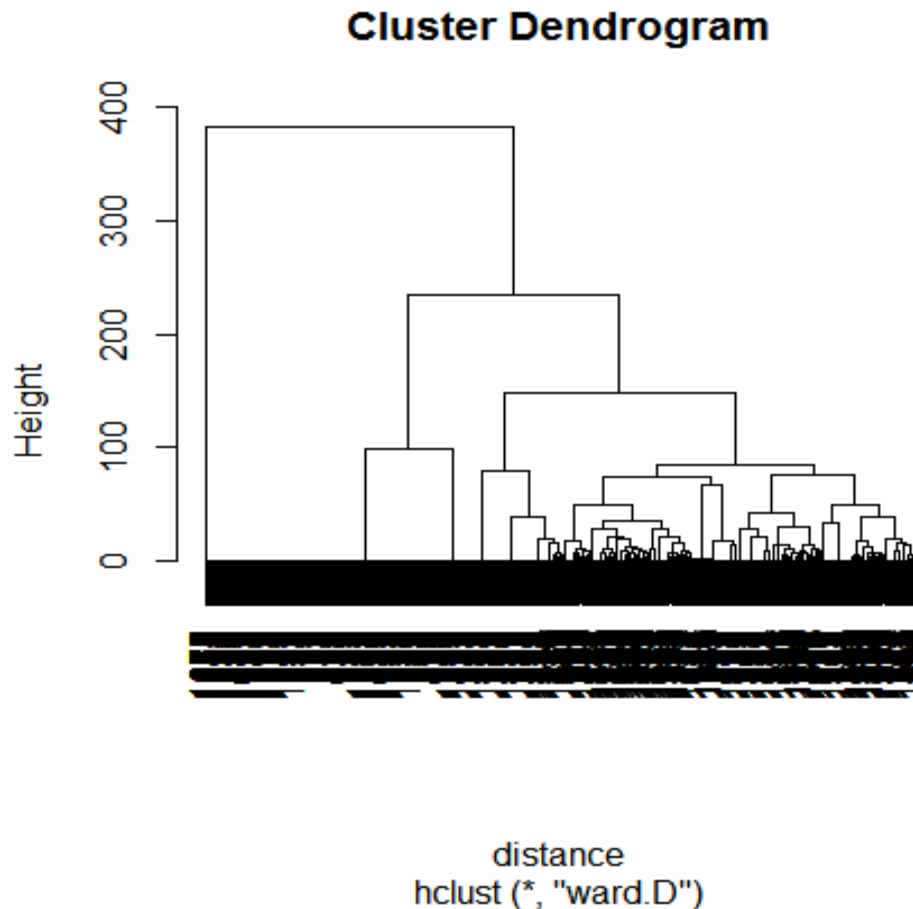
Clustering



- Now let's cluster our movies using the `hclust()` for hierarchical clustering
- We'll call the output `clusterMovies`
- The `hclust` takes the distance and method as arguments
- We will use the `ward.D` method.
- The ward method cares about the distance between clusters using centroid distance, and also the variance in each of the clusters.
 - `clusterMovies<- hclust(d = distance, method="ward.D")`

Dendrogram

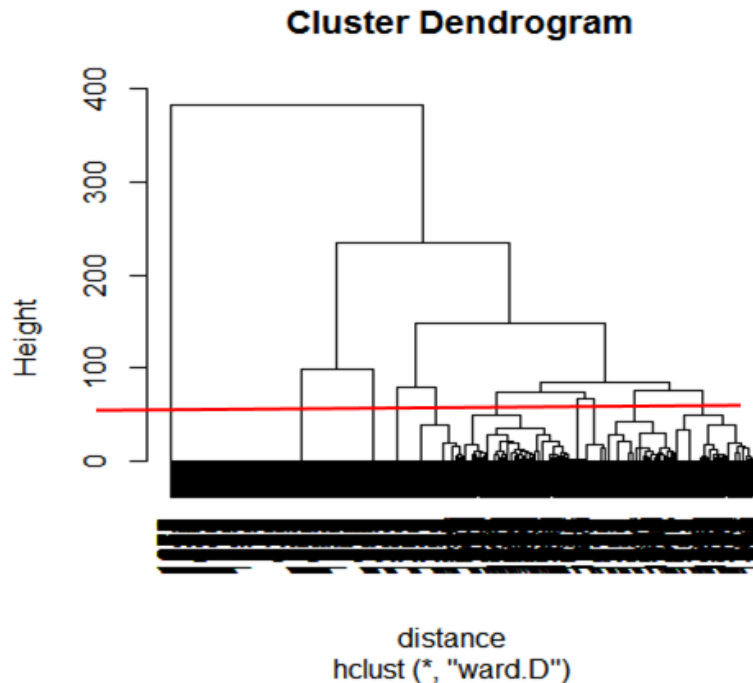
- `Plot(clusterMovies)` will give you the dendrogram.



Dendrogram

- The dendrogram lists all of the data points along the bottom.
- But when there are over 1,000 data points it's impossible to read.
- We have all this black along the bottom.
- So looking at this dendrogram, how many clusters would you pick?
- It looks like maybe three or four clusters would be a good choice.
- But we probably want more than two, three, or even four clusters of movies to make recommendations to users.

Dendrogram



- The red line is a good spot for us.
- We need to use our understanding of the problem to pick the number of clusters.
- Let's start with 10 clusters for now, combining what we learned from the dendrogram with our understanding of the problem.

Labeling of Data Points

- We need to label each of the data points according to what cluster it belongs
- This can be done using the `cutree()`.
 - `clusterGroups = cutree(clusterMovies,k=10)`
- We need to figure out what the clusters are like.
- We will use the `tapply` function to compute the percentage of movies in each genre and cluster.
- The action variable is a binary variable with value 0 or 1.
- So by computing the average of this variable we're computing the percentage of movies in each genre
 - `tapply(movies$Action,clusterGroups,mean)`

Labeling of Data Points

- Do it again for genre "romance"
– `tapply(movies$Romance, clusterGroup, s.mean)`
- Repeat the process for all parameters.

	A	B	C	D	E	F
1		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
2	Action	0.18	0.78	0.12	0.00	0.00
3	Adventure	0.19	0.35	0.04	0.00	0.00
4	Animation	0.13	0.01	0.00	0.00	0.00
5	Childrens	0.39	0.01	0.01	0.00	0.00
6	Comedy	0.36	0.07	0.06	0.00	1.00
7	Crime	0.03	0.01	0.41	0.00	0.00
8	Documentary	0.01	0.00	0.00	0.00	0.00
9	Drama	0.31	0.11	0.38	1.00	0.00
10	Fantasy	0.07	0.00	0.00	0.00	0.00
11	Film Noir	0.00	0.00	0.11	0.00	0.00
12	Horror	0.02	0.08	0.02	0.00	0.00
13	Musical	0.19	0.00	0.00	0.00	0.00
14	Mystery	0.00	0.00	0.28	0.00	0.00
15	Romance	0.10	0.05	0.04	0.00	0.00
16	Sci-Fi	0.07	0.35	0.04	0.00	0.00
17	Thriller	0.04	0.38	0.61	0.00	0.00
18	War	0.23	0.02	0.00	0.00	0.00
19	Western	0.09	0.00	0.00	0.00	0.00
20		Misc	Action-Adventure-SciFi	Crime-Mystery-Thriller	Drama	Comedy

	A	G	H	I	J	K	
1		Cluster 6	Cluster 7	Cluster 8	Cluster 9	Cluster 10	
2	Action	0.10	0.00	0.00	0.00	0.00	
3	Adventure	0.00	0.00	0.00	0.00	0.00	
4	Animation	0.00	0.00	0.00	0.00	0.00	
5	Childrens	0.00	0.00	0.00	0.00	0.00	
6	Comedy	0.11	1.00	0.02	1.00	0.16	
7	Crime	0.05	0.00	0.00	0.00	0.00	
8	Documentary	0.00	0.00	1.00	0.00	0.00	
9	Drama	0.66	0.00	0.00	1.00	0.00	
10	Fantasy	0.00	0.00	0.00	0.00	0.00	
11	Film Noir	0.01	0.00	0.00	0.00	0.00	
12	Horror	0.02	0.00	0.00	0.00	1.00	
13	Musical	0.00	0.00	0.00	0.00	0.00	
14	Mystery	0.00	0.00	0.00	0.00	0.00	
15	Romance	1.00	1.00	0.00	0.00	0.00	
16	Sci-Fi	0.00	0.00	0.00	0.00	0.00	
17	Thriller	0.14	0.00	0.00	0.00	0.16	
18	War	0.00	0.00	0.02	0.00	0.00	
19	Western	0.00	0.00	0.00	0.00	0.00	
20		Romance	Romantic Comedies	Documentary	Dramatic Comedies	Horror	
21							

Recommendation Systems

- Knowing common movie genres, these clusters seem to make a lot of sense.
- Amy liked the movie Men in Black.
- Figure out what cluster Men in Black is in?
- Make a subset of all the movies in that cluster.
- Look at the first 10 titles in this cluster.
- Good movies to recommend to Amy, according to our clustering algorithm, would be movies like Apollo 13 and Jurassic Park.

Quick Question

- Run the `cutree` function again to create the cluster groups, but this time pick $k = 2$ clusters. It turns out that the algorithm groups all of the movies that only belong to one specific genre in one cluster (cluster 2), and puts all of the other movies in the other cluster (cluster 1). What is the genre that all of the movies in cluster 2 belong to?

☐ Crime☐ Documentary☐ Drama☐ Fantasy☐ Film Noir

The Analytics Edge of Recommendation Systems

Beyond Movies: Mass Personalization

- “If I have 3 million customers on the web, I should have 3 million stores on the web”
– Jeff Bezos, CEO of Amazon.com
- Recommendation systems build models about users’ preferences to personalize the user experience
- Help users find items they might not have searched for:
 - A new favorite band
 - An old friend who uses the same social media network
 - A book or song they are likely to enjoy

Cornerstone of these Top Businesses

www.dimensionless.in

amazon.com
and you're done.™

NETFLIX

last.fm



Spotify®



PANDORA

internet radio

Recommendation Method Used

- Collaborative Filtering
 - Amazon.com
 - Last.fm
 - Spotify
 - Facebook
 - LinkedIn
 - Google News
 - MySpace
 - **Netflix**
- Content Filtering
 - Pandora
 - IMDB
 - Rotten Tomatoes
 - Jinni
 - Rovi Corporation
 - See This Next
 - MovieLens
 - **Netflix**

The Netflix Prize: The Final 30 Days

www.dimensionsless.in

- 29 days after last call was announced, on July 25, 2009, the team The Ensemble submitted a 10.09% improvement
- When Netflix stopped accepting submissions the next day, BellKor's Pragmatic Chaos had submitted a 10.09% improvement solution and The Ensemble had submitted a 10.10% improvement solution
- Netflix would now test the algorithms on a private test set and announce the winners

Winners are Declared!

- On September 18, 2009, a winning team was announced
- BellKor's Pragmatic Chaos won the competition and the \$1,000,000 grand prize



The Edge of Recommendation Systems

- In today's digital age, businesses often have hundreds of thousands of items to offer their customers
- Excellent recommendation systems can make or break these businesses
- Clustering algorithms, which are tailored to find similar customers or similar items, form the backbone of many of these recommendation systems