Recommender Systems

Noopur Khachane
Mihir Sathe
Liyang Yu
Heguang Zhou
Jingpei Lu

What is a Recommender System?

- Given data about users and items, predict a user's preferences or ratings for items

Where do we see them today?

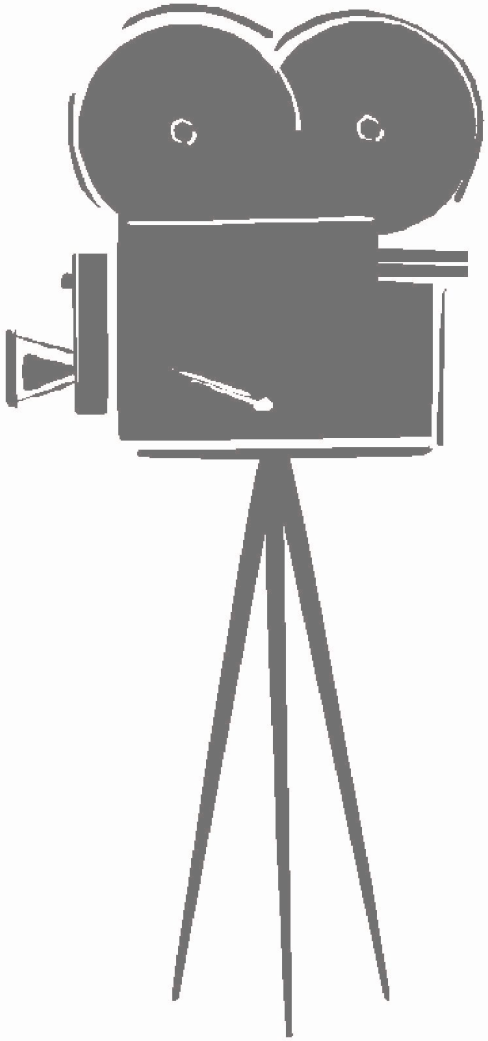
- Recommender systems are so common nowadays that we probably don’t even think about them.
  - Amazon’s “Suggested Items”
  - Spotify “Discover Playlist”
  - Netflix’s “Match Score”

Netflix Prize

- From the years 2006-2009, Netflix offered up a prize of 1 million dollars to a team that could beat their recommender system by 10%
- The winners used combinations of many algorithms, one team even used about 107.
- The challenge was to predict without information about users

MovieLens100K

- Website that supplies movie recommendations
- Open-Source dataset
- Includes (100,000)
  - User Info(Age, Gender, ZIP, Occupation)
  - Movie Info(Release date, Genre)



Collaborative Filtering

- Find ‘Neighbors’ with similar preferences to target user
- Use a similarity weighted average to predict
- Eg. K-Nearest Neighbors

Content-Based Filtering

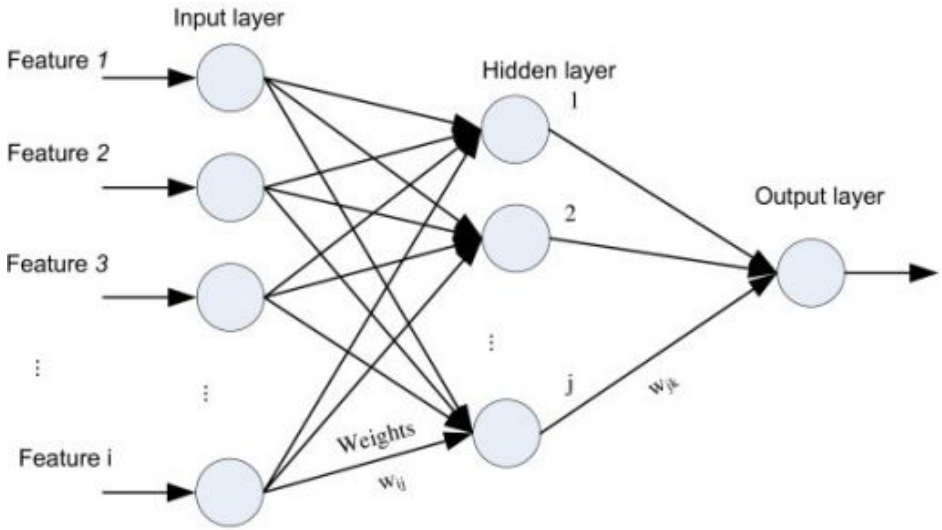
- Weight all movies with respect to similarity with the active movie.
- Select a subset of the movies(*neighbors*) to use as predictors.
- Compute a prediction from a weighted combination of the selected neighbors' ratings.

Hybrid Systems

- Take information from Collaborative/Content Filtering
- Compute a prediction from a weighted combination of the two models

Neural Network

- Engineer Features based on (Users, Movies, Ratings)
- Train Network on (Features, Ratings)
- Predict for new (User,Movie)



Process Data First!

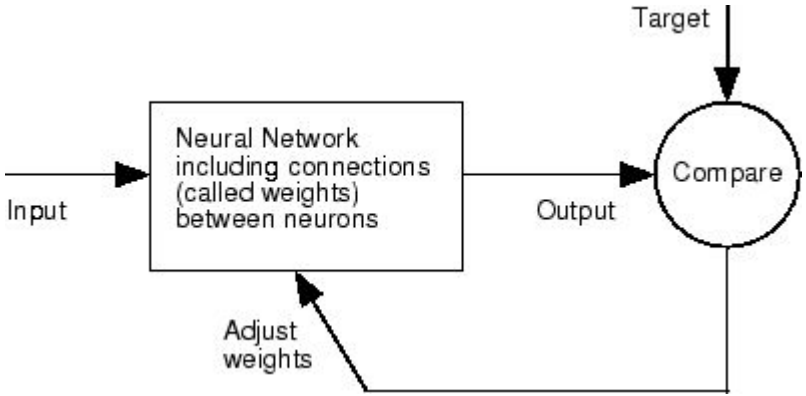
- Split into Training/Test with 80:20 Ratio
- 80,000 Training Examples, 20,000 Test Examples

Neural Network

- Engineer Features based on (Users, Movies, Ratings)
  - three_year_mean(Users,Movies,Ratings)
  - gender_mean(Users,Movies,Ratings)
  - user_mean(User,Ratings)
  - movie_mean(Movie,Ratings)

Neural Network

- Train on 80,000 (Features, Ratings)
  - The network learns to predict a rating based on a function of its features



Neural Network

- Predict on 20,000 Test Points
- Calculate Root Mean Squared Error = 0.85★
  - “How many stars are we ‘off’ by?”
- Calculate Explained Variance = 67.5%
  - “What percent of the variation in our dataset does our model explain?”

Future Improvements

- Predict on more data (MovieLens1M)
- Extract more features
  - We threw away ZIP, Occupation
- Optimize Model Parameters
  - How to choose size of network? Learning Rate? etc.

Thank You!