Singular Value Decomposition (SVD) to Recommend Movies

By: Cole Hanniwell, Einstein Oyewole, Jack Pentesco and Jacob Orrico

Agenda

- Introduction
- Dataset
- First Model
- Our Model
- Experiments
- Evaluation and Results
- Conclusion

Introduction

- **GOAL:** develop a movie recommendation system
- **OBJECTIVE:** accurately predict ratings that a user would give a movie
- **HOW:** recommend movies with higher predicted ratings to users

Dataset

What data did you use?

- Dataset of 610 users with 100386 ratings across 9742 movies
- Average overall rating of 3.5 with a standard deviation of 1.04
- First movie rated in 1996 and the last movie was rated in 2018
- 1475 unique tags that users gave to describe the movies
- Most common tag is “In Netflix Queue”
- Most common genres are “no genre listed” and “Film-Noir”

	userId	movieId	tag	timestamp
0	2	60756	funny	1445714994
1	2	60756	Highly quotable	1445714996
2	2	60756	will ferrell	1445714992
3	2	89774	Boxing story	1445715207
4	2	89774	MMA	1445715200

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

Model

First Model

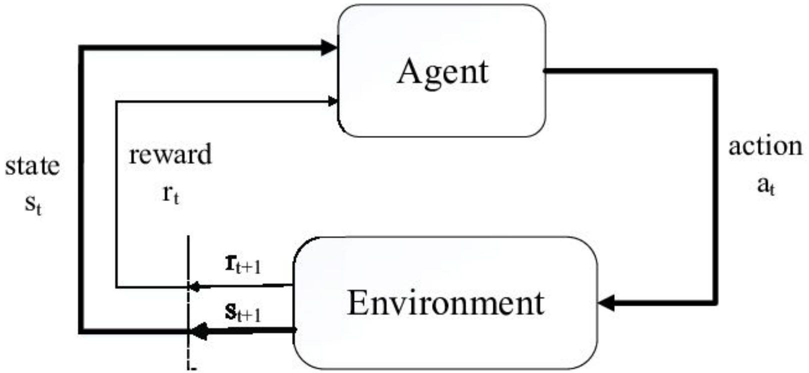


Diagram Overview

- The action represents the movies they recommend
- Rewards in the diagram represent the ratings
- The state is the movies users have watched

Reinforced Learning (value-based)

- Goal is to maximize the immediate reward
- Actions and rewards must be trained, so that the model can generate new actions

Exploitation vs. Exploration

- Exploitation refers to the optimal actions
- Exploration aims to find a better solution
- Recommend actions and popular movies, helps us learn more about the user

First Model

Steps:

1. Dimension reduction (make sure to capture 80% of variance)
2. Similarity = Euclidean Distance, Cosine Similarity, Minkowski @p = 5
3. Build the Q matrix , Q-value Score = [(rating - threshold)/threshold]*similarity
4. Use the ratings as rewards and propagate it in the users state
5. Threshold = 2.5
6. Recommend movies to users

First Model

Why did we choose this method?

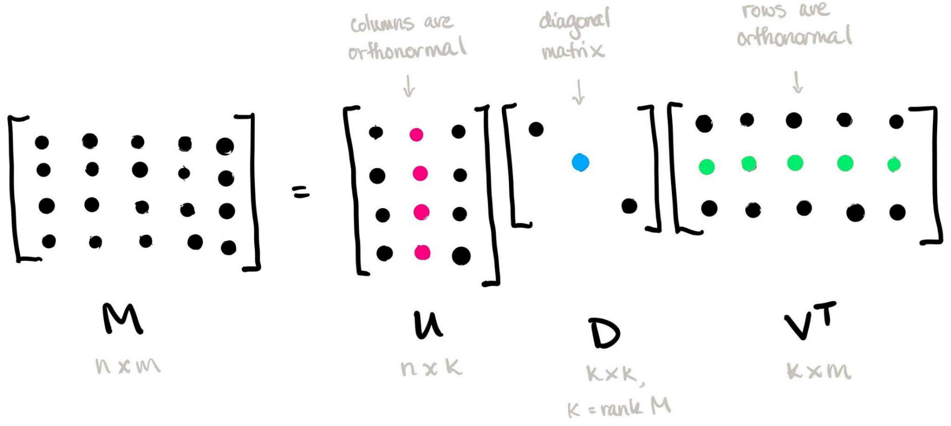
- Try to avoid limitations of collaborative filter and content based
  - These limitations are:
    - Difficult to make recommendations for new users and overspecialization for content based
    - Difficult to include other features and hard to predict ratings for brand new movies for collaborative filtering

Why it failed:

- Only suggested movies for users and did not predict actual ratings
- Could not use much training data because the computation time was too long
- When attempting to convert suggestions to ratings the model did not perform well

Our Model

- SVD is a collaborative filtering method
- SVD does a factorization of the matrix with UserId, MovieId and Rating into three separate matrices
- These 3 matrices represent low dimensional hidden factors (characteristics) for users and movies
- Use these matrices to predict ratings that did not exist in original matrix



Our Model

Steps:

1. Compute transpose of original matrix and multiply transpose matrix by the original matrix
2. Determine eigenvalues of the this matrix and sort them in descending order and square root to obtain singular values of the original matrix
3. Create matrix where the diagonal values are the singular values sorted in descending order and find the inverse of this matrix (Matrix D)
4. Use eigenvalues from step 2 to find the eigenvectors and place these along the columns of a new matrix and find the transpose of this matrix (Matrix V)
5. Using the original matrix, S and V solve for Matrix U

Experiments

- Take a test set out of the data
- Predict ratings for movies the users have already rated and compare the predicted ratings to the actual ratings to evaluate model
- Root Mean-Square Error (RMSE) is the metric
- Comparison methods (these are different prediction methods):
- Use global mean of all ratings, use average rating of each user, use average rating of each movie, classic methods (NMF and KNN)
- Compare all these to our Model RMSE

Evaluation and Results

- Global mean RMSE = 150.60
- User Mean RMSE = 132.37
- Movie Mean RMSE = 112.06
- Classic Method RMSE = NMF = 0.9220, KNN = 0.9477
- Our Model RMSE (SVD) = 0.8489

Demo

Conclusion

- Our model performs significantly better than the other prediction methods
- This means our ratings are more accurate
- To further enhance the model we could add more data

References

- SVD & NMF https://surprise.readthedocs.io/en/stable/matrix_factorization.html
- KNN https://surprise.readthedocs.io/en/stable/knn_inspired.html
- Active RL *Russell, Stuart J. (Stuart Jonathan). Artificial Intelligence : a Modern Approach. Upper Saddle River, N.J. :Prentice Hall, 2010*

Questions?