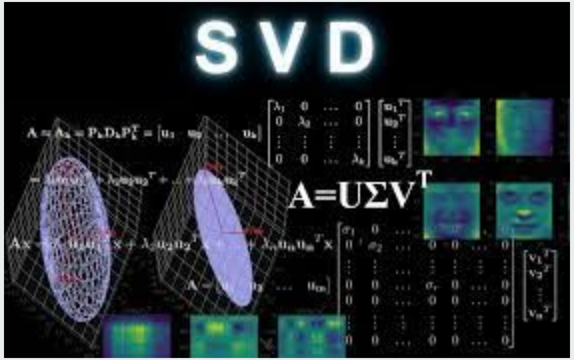
Collaborative filtering - Singular value decomposition project

CMPE 297-Special Topics

Professor: Dr. Gheorghi Guzun



Group : 9
Alireza Mahinparvar
Roopa Chiluvuri
Shanmukh Krishna
Yashwanth Kumar Nelakuditi

Introduction

Recommendation Systems are information filtering systems that provide suggestions for items (movies in our scenario) that are most pertinent to a particular user.

Problem Statement : To recommend a movie to an user , we calculate the similarities between the movies using similarity measures like cosine or pearson's correlation. While doing this, we are facing issues like scalability and sparsity.

To overcome such problems, we came up with **SVD** and **SVD++** models where they help us better to understand the relationship between users and movies as they are directly comparable.

Dataset

The “ratings” dataset which we have used in our project is from Kaggle. Due to the limited computing capacity, we have used a shortened version of the dataset which named as ratings_small.csv

1. userid - a unique identifier for each user
2. movieid - a unique identifier for each movie
3. rating - rating of a movie given by a user
4. timestamp - time at which this rating has been given

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

Singular Value Decomposition

One way to handle these issues is to leverage a latent factor model to find similarities. **SVD** decreases the dimension of the utility matrix by extracting its latent factors. We map each user and each movie into a latent space with dimension r where they become directly comparable.

Singular Value Decomposition of an mxn complex matrix M is a factorization of the form

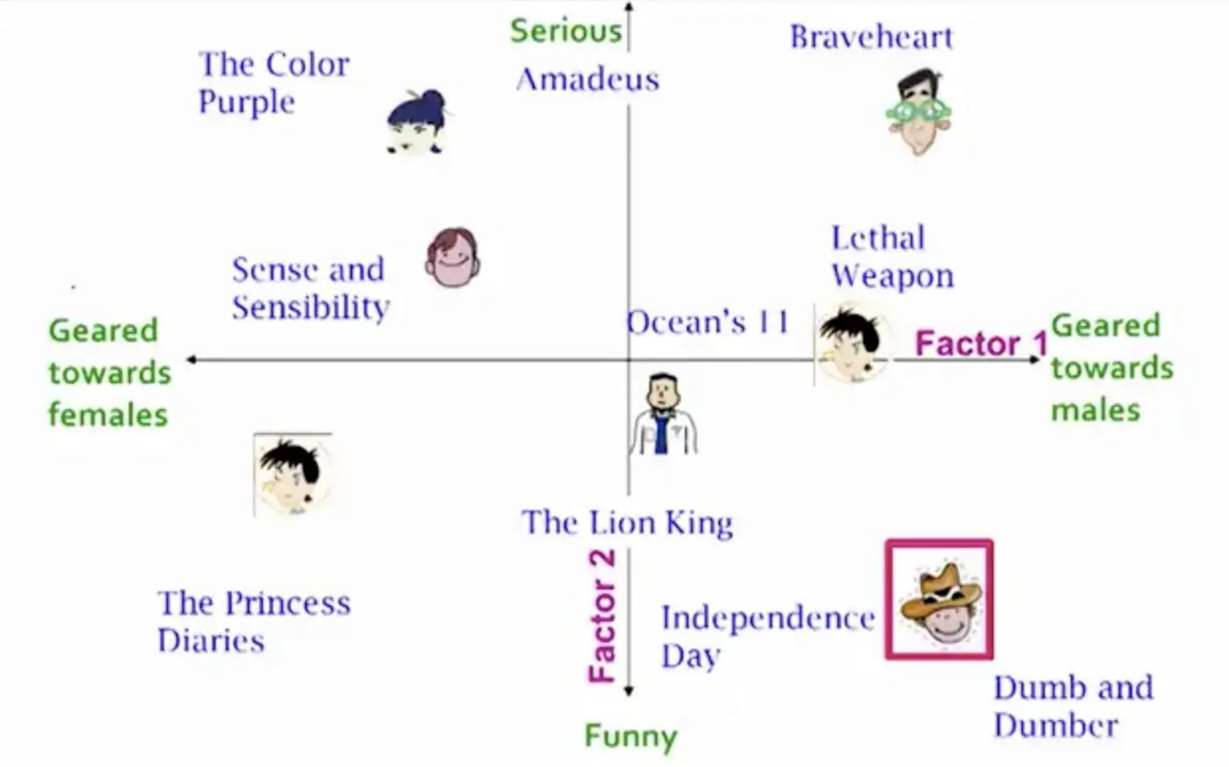
$$\mathbf{M = U\Sigma V^T}$$

Where U is mxm complex unitary matrix

Σ is mxn rectangular diagonal matrix

V^T is conjugate transpose of V

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$


Serious
Amadeus
Braveheart
The Color Purple
Sense and Sensibility
Lethal Weapon
Ocean's 11
Factor 1
Geared towards females
Geared towards males
The Lion King
Factor 2
Independence Day
The Princess Diaries
Dumb and Dumber
Funny

SVD++

SVD++ is an extended version of SVD where it improves precision and recall of the recommender system.

"++" means incorporating implicit feedback.

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

Where the y_j terms are a new set of item factors that capture implicit ratings. Here, an implicit rating describes the fact that a user u rated an item j, regardless of the rating value.

Hyper Parameter Tuning

To find the optimal parameter values from a given set of parameters in a grid, we used a cross validation technique, GridSearchCV.

METRICS :

Root Mean Square Error (RMSE) : It is frequently used to measure the differences between values predicted by a model and the values observed.

Mean Absolute Error (MAE) : It measures the difference between the measured values and true values.

We used **RMSE** and **MAE** to measure the performance of the modes

Code Walkthrough

RMSE and MAE Scores

	RMSE	MAE	Fit Time	Best Time
SVD	0.8975	0.6905	0.87	0.12
SVD++	0.8863	0.6793	47.73	6.88
Hyper parameter Tuning (SVD)	0.8902	0.6868	0.47	0.16
Hyper parameter Tuning (SVD++)	**0.8856**	0.6821	96.19	6.02

Results

	SVD	Hyper parameter tuning	SVD++	Hyper parameter tuning
Actual Ratings	2.5	2.5	2.5	2.5
Predict Ratings	2.3445	2.4324	2.4407	**2.5076**

Other algorithms

We also tried other algorithms such as KNNWithMeans, NMF, BaselineOnly etc.,

Algorithm	RMSE	MAE
SVDpp	0.8954	0.6873
BaselineOnly	0.8962	0.6932
SVD	0.9018	0.6951
KNNWithMeans	0.9291	0.7103
NMF	0.9606	0.7378
KNNBasic	0.9795	0.7533

Conclusion

- In conclusion, we have evaluated results by implementing various models such as **SVD** and **SVD++** along with hyper parameter tuning to find the best predictor among them.

Future Scope

- SVD is a very popular technique that can be found in other libraries as well. So we can use the SVD present in other libraries and perform the same operations and can compare the results obtained by other SVD's.
- The input used in this project is a dense matrix. This can be converted into a sparse matrix and perform the predictions and observe the results obtained.

Thank you !!

Q/A