Recommendation Systems with Matrix Factorization

Onur Poyraz, Serkan Emre Elçi, Sinan Selamoğlu

Abstract

Recommendation systems are used in various applications, especially in e-commerce, such as recommendation of musics, books and movies. Recommender systems uses content-based filtering which may require collecting extra information from users and collaborative filtering which can analyze relationships between users and items by characterizing data with latent factor models. In this project, we compare different matrix factorization models such as Probabilistic Matrix Factorization, Matrix Factorization, Nonnegative Matrix Factorization, Logistic Matrix Factorization and Alternating Least Square on sparse and large movielens dataset as a collaborative filtering approach.

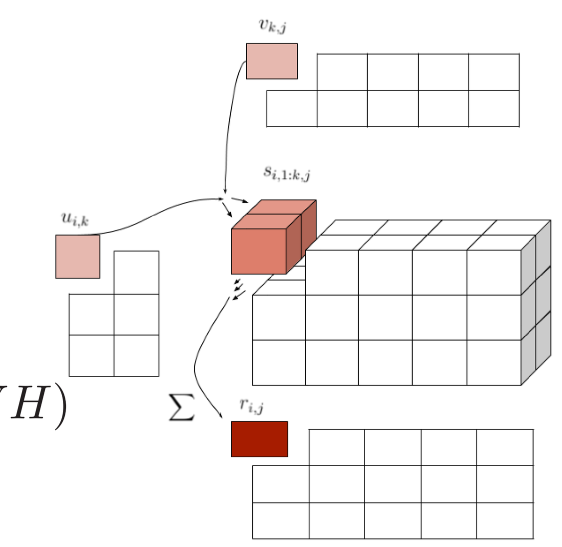
Introduction

The aim of this project is to implement different matrix factorization models on MovieLens dataset and observe the results. In this project, we aim to see the effect of the sparsity of the data and the different kind of matrix factorization techniques. In movielens dataset 100K, since all users could not able to rate all movies, matrix factorization methods plays a crucial role as collaborative filtering to predict missing rating. Even through dataset used is the same for each model, the error results can be different. The experiment will show us which matrix factorization model is more suitable for a recommendation system.

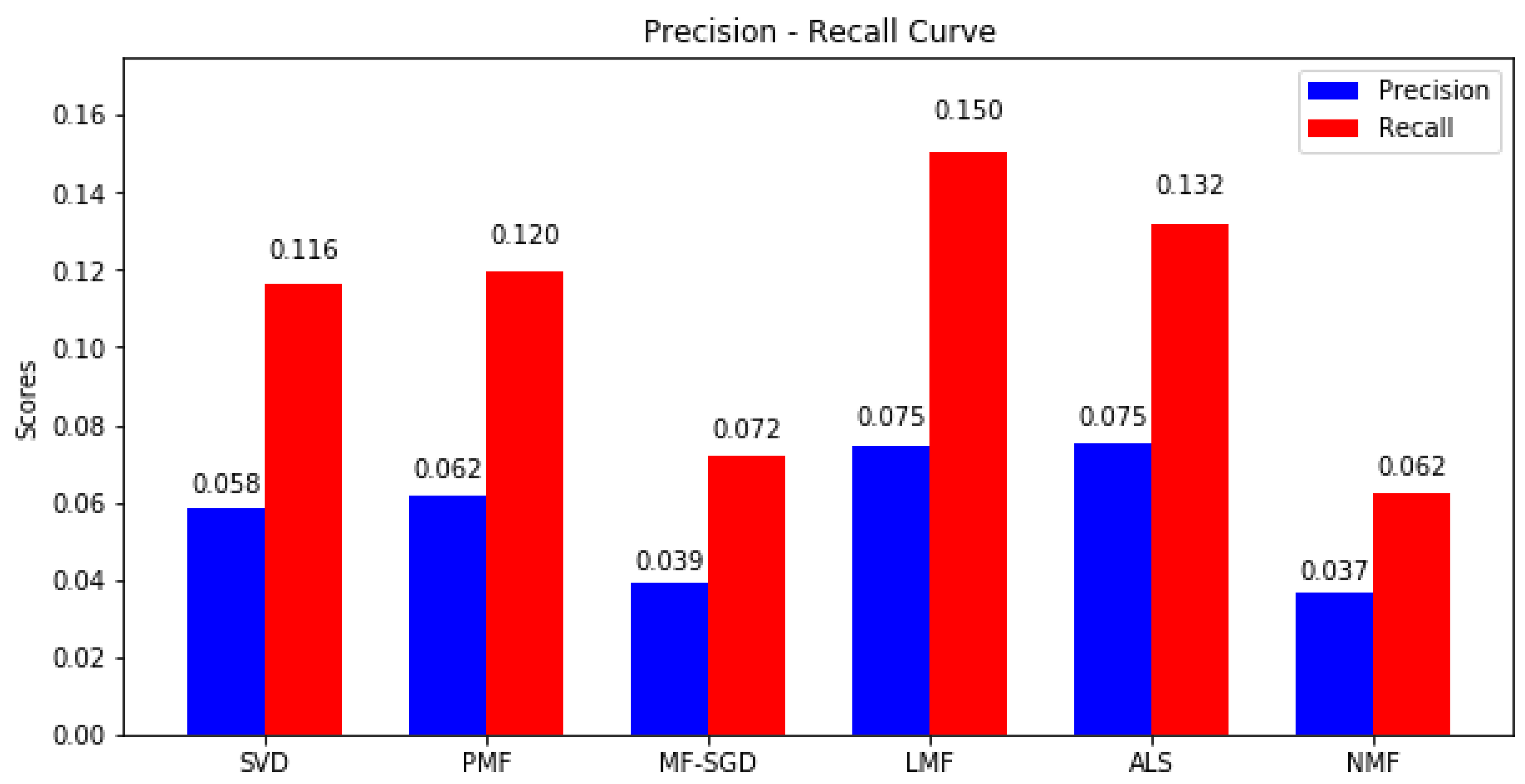
Problem

1 What is Matrix Factorization?

- Input $R = \begin{bmatrix} r_{11} & \dots & r_{1J} \\ \dots & \dots & \dots \\ r_{I1} & \dots & r_{IJ} \end{bmatrix}$
- Approach: $R \approx WH$
- Obj: $(W,H)^* = argmin_{W,H} D(R \| WH)$
- $R_{ij} \approx r_{ij} = \sum_k s_{ikj}$



Evaluation



Derivations

Singular Value Decomposition

$$\hat{R} = R - b - b_i - b_u$$
$$\hat{R} = U \Sigma V^T$$

Logistic Matrix Factorization

$$\begin{aligned}
\mathcal{L}(W,H) &= \sum_j \sum_i M(i,j) Y(i,j) \big(\sum_k W(i,k) H(k,j)\big) \\
&- \sum_j \sum_i M(i,j) \log \Big(1 + \exp\big(\sum_k W(i,k)H(k,j)\big)\Big) \\
\frac{\partial \mathcal{L}(W,H)}{\partial W(i,k)} &= \sum_j M(i,j) \left(Y(i,j) - \sigma \left(\sum_k W(i,k) H(k,j) \right) \right) H(k,j) \\
\frac{\partial \mathcal{L}(W,H)}{\partial H(k,j)} &= \sum_j M(i,j) \left(Y(i,j) - \sigma \left(\sum_k W(i,k) H(k,j) \right) \right) W(i,k)
\end{aligned}$$

Matrix Factorization with SGD

$$\begin{aligned}
E(W,H) &= \frac{1}{2} \sum_i \sum_j M(i,j) \Big(X(i,j) - \sum_k W(i,k) H(k,j) \Big)^2 \\
\frac{\partial E(W,H)}{\partial W_{i,k}} &= -\sum_j M(i,j) \Big(X(i,j) - \sum_{k'} W(i,k') H(k',j) \Big) H(k,j) \\
\frac{\partial E(W,H)}{\partial H_{k,j}} &= -\sum_i M(i,j) \Big(X(i,j) - \sum_{k'} W(i,k') H(k',j) \Big) W(i,k)
\end{aligned}$$

Probabilistic Matrix Factorization

$$\begin{aligned}
E(W,H) &= \frac{1}{2} \sum_i \sum_j M(i,j) \Big(X(i,j) - \sum_k W(i,k) H(k,j) \Big)^2 \\
&+ \frac{\lambda_W}{2} \sum_i \|W_i\|_F^2 + \frac{\lambda_H}{2} \sum_j \|H_j\|_F^2
\end{aligned}$$

Nonnegative Matrix Factorization (Beta (KL) Divergence)

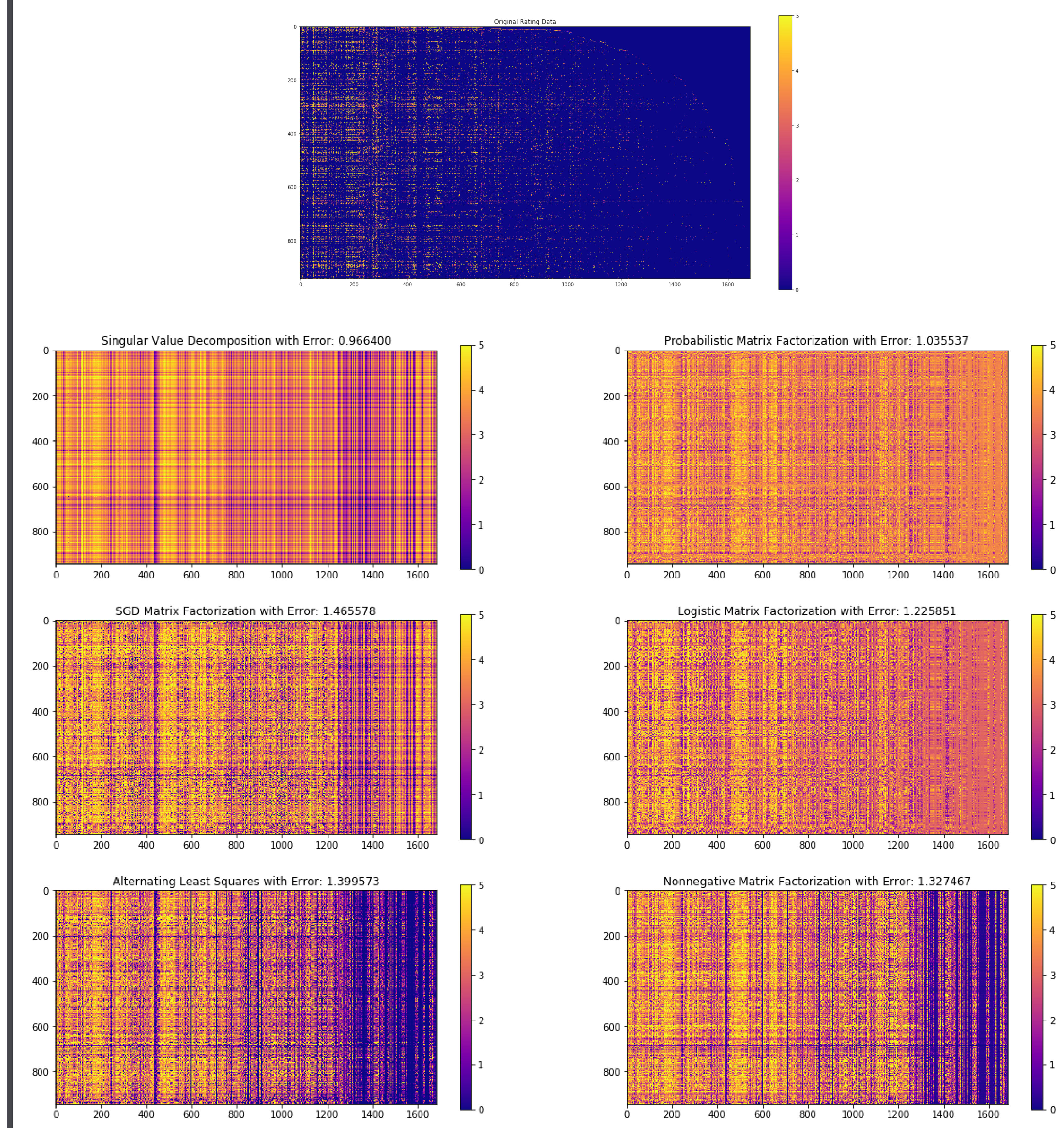
$$\frac{\partial D(R; \hat{R})}{\partial W(i,k)} = \sum_i \frac{\partial d_p(R(i,j); \hat{R}(i,j))}{\partial \hat{R}(i,j)} \frac{\partial \hat{R}(i,j)}{\partial W(i,k)}$$

$$W(i,k) \leftarrow W(i,k) \frac{\sum_j H(k,j) R(i,j) / \hat{R}(i,j)^p}{\sum_j H(k,j) \hat{R}(i,j)^{1-p}}$$

Alternating Least Squares

$$\begin{aligned}
\|E\|_F^2 &= Tr\left(R^\top R + H^\top W^\top W H - 2R^\top W H\right) \\
H &= (W^\top W)^{-1} W^\top R \\
W &= R H^\top (H H^\top)^{-1}
\end{aligned}$$

Results



Conclusion

In this project, we implemented SVD, Probabilistic Matrix Factorization, Matrix Factorization with SGD, Nonnegative Matrix Factorization, Alternating Least Square and Logistic Matrix Factorization on Movielens dataset. For healthy comparison, we make rank 10 approximation on all models. We split 50 percent of data is used for training and other half for testing. Each model produces a different kind of results because of their types and error metrics they used. Our result shows that Logistic Matrix Factorization performs better among other models with respect to precision and recall rate.

Reference

1. Salakhutdinov et.al., Probabilistic Matrix Factorization
2. Koren et al., Matrix Factorization Techniques for Recommender Systems
3. Fevotte et al.,Algorithms for Nonnegative Matrix Factorization with the β-Divergence
4. Taylan Cemgil, Tutorial on Coupled Tensor Decomposition
5. $https://github.com/CMPE462-Spring2018-Bogazici/term-project-onur$