Recommender System Based on Matrix Factorization with Incorporation of Movie Genre Information

CAI Bibi
Department of Mathematics
HKUST
bcaiaa@connect.ust.hk

QIU Zhenyu
Department of Mathematics
HKUST
zqiuai@connect.ust.hk

WANG Zhiwei
Department of Mathematics
HKUST
zhiwei.wang@connect.ust.hk

Abstract

In this report, we developed a recommendation system based on the probabilistic matrix factorization framework with the incorporation of movie genre information and analyzed the MovieLens 100K data. Specifically, the main data matrix can be factorized as two low-dimension matrices by maximizing the evidence lower bound of the logarithm of the marginal likelihood. Then, the two latent factor matrices can respectively represent the scores of movies and the preferences of users. Our method can also handle missing data and impute the users' ratings on movies and recommend movies to users. We examined the prediction accuracy of the proposed method by comparing the testing mean absolute errors (MAEs) with existing methods, such as Funk SVD, SoftImpute, Truncated SVD, and KNN Imputer. Our method can also explicitly show the connection between the latent movie factors and genre information, which helps gain deeper insight. In addition, we further explored the relationship between the preferences of users and users' features and visualized the results via the data reduction and clustering methods. Our results showed that the proposed model enjoys high prediction accuracy and the scores of movies on latent factors can be mostly explained by a linear combination of movies' genres, which indicates that our model can well cope with the movie-level cold-start problem.

1 Introduction

MovieLens 100K Dataset consists of a main matrix containing users' ratings and side information about corresponding movies' genres and users' features, such as age and occupation. The main data matrix is in the form of a 1682 * 943 matrix of users' ratings, recording 943 users' ratings on 1682 movies. Each row contains the ratings of users on each movie. Since only a few users have viewed each movie, the main matrix is extremely sparse and has only 100,000 ratings in total. The other elements in the matrix remain null, which can be viewed as missing values (ratings). Also, side information for movies' genres represents the genres with dummy variables. Note that each movie can own more than one genre. Additionally, side information for users' features is a table illustrating the age, gender, and occupation of every user.

In order to predict the missing ratings, we propose a matrix factorization model which can decompose the users' rating matrix while incorporating the information on genres of movies. After the decomposition process in an expectation maximization (EM) framework, we can estimate and recover the original main matrix by multiplication of two decomposed matrices and meantime fill in the missing ratings. The predicted ratings can be used to further recommend movies to users. We implement the proposed model and perform an experiment to examine the prediction power of the proposed model by comparing the testing mean absolute errors (MAEs) with existing methods, such as Funk SVD, SoftImpute, Truncated SVD, and KNN Imputer.

Submitted to CSIC5011 Topological and Geometric Data Reduction and Visualization, 2023 Spring, HKUST.

Next, the model decomposes the main data matrix into two low-dimensional matrices, which respectively represent the scores of movies and preferences of users on different latent factors. We explore how the scores of movies on latent factors relate to the genres of movies by performing linear regression and statistically analyzing the regression performance and coefficients.

In addition, we view the decomposed matrix represented preferences of users on latent factors as a kind of embedding of users. Then, we conduct dimension reduction methods to find the lower dimensional embedding for users' preferences on movies and further explore its relationship with users' features by visualization.

2 Research Questions

- Can we further improve the imputation accuracy by leveraging the side information based on the existing matrix factorization framework?
- How do the latent factors relate to the side information based on the above model? Can we gain deeper insight into the data?

3 Proposed Model

Given the main data matrix $\mathbf{Y} \in \mathbb{R}^{N \times M}$ of N samples and M features, we consider the following matrix factorization problem

$$\mathbf{Y} = \mathbf{Z}\mathbf{W}^{\mathrm{T}} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times K}$ and $\mathbf{W} \in \mathbb{R}^{M \times K}$ are two matrices, and $\boldsymbol{\epsilon} \in \mathbb{R}^{N \times M}$ is a matrix of residual error terms. Here we adopt the terminology of factor analysis and refer to $\mathbf{Z}$ as the "loadings", $\mathbf{W}$ as the "factors", and K as the number of factors. We can further expand the above formulation as the sum of the K factors

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{Z}_{\cdot k} \mathbf{W}_{\cdot k}^{\mathrm{T}} + \boldsymbol{\epsilon}, \tag{2}$$

where $\mathbf{Z}_{\cdot k}$ and $\mathbf{W}_{\cdot k}$ are the k-th column of $\mathbf{Z}$ and $\mathbf{W}$, respectively. Let's take a user-movie rating matrix $\mathbf{Y}$ as an example, where the observed entry $\mathbf{Y}_{nm}$ represents the rating score of user m for movie n. We may assume that the K factors represent the K traits of movies, where $\mathbf{W}_{\cdot k}$ corresponds to the preferences of M users related to the k-th trait (e.g., action or emotional), $\mathbf{Z}_{\cdot k}$ corresponds to the scores of N movies on the k-th trait. The product of preference and score on the k-th trait measures the strength of the k-th factor. The final rating depends on the overall effects of K factors. In the probabilistic PCA framework, we often assume the simplest case, i.e., independent loadings and factors. Therefore, we then assign independent Gaussian priors for the k-th pair of loading/factor $\mathbf{Z}_{\cdot k}$ and $\mathbf{W}_{\cdot k}$

$$\mathbf{Z}_{\cdot k} \sim \mathcal{N}_N(0, \beta \mathbf{I}_N), \tag{3}$$

where β is a scaling parameter, and

$$\mathbf{W}_{\cdot k} \sim \mathcal{N}_M(0, \mathbf{I}_M), \tag{4}$$

and for independent error terms $\boldsymbol{\epsilon}$

$$\epsilon_{nm} \sim \mathcal{N}(0, \tau^{-1}),\ n = 1, \ldots, N \text{ and } m = 1, \ldots, M, \tag{5}$$

where τ is the precision of ϵ_{nm}. In the statistical machine learning literature, $\mathbf{Z}$ and $\mathbf{W}$ are often referred to as latent variables.

To perform matrix factorization of $\mathbf{Y}$, we not only have observed entries in the main matrix but also some side information. For example, we often have movie information $\mathbf{X} \in \mathbb{R}^{N \times C}$ in the above movie rating case, where each row represents a movie belonging to C genres, such as action, adventure, animation, and so on. We incorporate side information into matrix factorization by assuming that the movies' factors are associated with movie genres. Typically, we have two ways to directly incorporate this kind of association in our model. The first way is to model the mean of $\mathbf{Z}$

$$\mathbf{Z}_{\cdot k} \sim \mathcal{N}_N\left(F_k\left(\mathbf{X}\right), \beta_k^{-1} \mathbf{I}_N\right),\ k = 1, \ldots, K, \tag{6}$$

where $F_k(\mathbf{X}) \in \mathbb{R}^{N \times 1}$ is the mean vector of the loading $\mathbf{Z}_{\cdot k}$, β_k is the precision, $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ is an identity matrix, and $\mathcal{N}_N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the N-variate Gaussian distribution with mean $\boldsymbol{\mu}$ and

variance $\mathbf{\Sigma}$. Note that $F_k(\mathbf{X})$ is the row-wise evaluation of the unknown function $F_k : \mathbb{R}^C \to \mathbb{R}$, $F_k(\mathbf{X}) = (F_k(\mathbf{X}_{1\cdot}), \ldots, F_k(\mathbf{X}_{N\cdot}))^{\mathrm{T}}$, where $\mathbf{X}_{n\cdot} = (\mathbf{X}_{n1}, \ldots, \mathbf{X}_{nC})^{\mathrm{T}} \in \mathbb{R}^{C\times 1}$ is the n-th row of $\mathbf{X}$ containing side information for the n-th sample, $n = 1, \ldots, N$. Alternatively, we can also turn to model the covariance matrix. We introduce the following automatic relevance determination (ARD) Gaussian process prior

$$\mathbf{Z} \sim \mathcal{N}_N(0, \alpha\mathbf{K} + \beta\mathbf{I}_N), \tag{7}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ and specifically,

$$\mathbf{K}_{ij} = k\left(\mathbf{X}_i, \mathbf{X}_j\right) \tag{8}$$

where k is the kernel function, $\mathbf{I}_N$ is the $N \times N$ identity matrix, α and β are two ARD parameters, and $\mathcal{N}_N(\boldsymbol{\mu}, \mathbf{\Sigma})$ denotes the N-variate Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\mathbf{\Sigma}$. One commonly used kernel is the squared exponential kernel

$$\mathbf{K}_{ij} = \exp\left(-\frac{||\mathbf{X}_i - \mathbf{X}_j||_2^2}{\gamma}\right) \tag{9}$$

where γ is the length scale parameter. We can also combine these two modeling ways together

$$\mathbf{Z} \sim \mathcal{N}_N(F_k\left(\mathbf{X}\right), \alpha\mathbf{K} + \beta\mathbf{I}_N). \tag{10}$$

However, considering the covariance matrix in our model suffers from cubic computation cost since the fitting algorithm requires computing the matrix inverse in the iteration. Therefore, we choose to model the loading mean and consider the simplest case, i.e., linear function. Specifically, we relate $\mathbf{Z}_{\cdot k}$ and covariates $\mathbf{X}$ using the following probabilistic model

$$F_k(\mathbf{X}_{n\cdot}) = \gamma_{0k} + \mathbf{X}_{n\cdot}^{\mathrm{T}}\boldsymbol{\gamma}_{\cdot k}, \tag{11}$$

where γ_{0k} is the intercept and $\boldsymbol{\gamma}_{\cdot k} \in \mathbb{R}^{C\times 1}$ represents regression coefficients.

Let $\boldsymbol{\theta} = \{\tau, \boldsymbol{\beta}, \boldsymbol{\gamma}\} = \{\tau; \beta_1, \ldots, \beta_K; \boldsymbol{\gamma}_{\cdot 1}, \ldots, \boldsymbol{\gamma}_{\cdot K}\}$ be the collection of model parameters. Combining model (2) (6) (4) (5) (11) and considering the missing entries in the observed data matrix, we can write down the joint probabilistic model as

$$\begin{aligned}
\Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\theta}\right) &= \Pr\left(\mathbf{Y}^{\text{obs}} \mid \mathbf{Z}, \mathbf{W}; \tau\right) \Pr\left(\mathbf{Z} \mid \boldsymbol{\beta}, \boldsymbol{\gamma}\right) \Pr\left(\mathbf{W}\right) \\
&= \Pr\left(\mathbf{Y}^{\text{obs}} \mid \mathbf{Z}, \mathbf{W}; \tau\right) \prod_{k=1}^{K} \Pr\left(\mathbf{Z}_{\cdot k} \mid \beta_k, \boldsymbol{\gamma}_{\cdot k}\right) \prod_{k=1}^{K} \Pr\left(\mathbf{W}_{\cdot k}\right) \\
&= \prod_{(n,m)\in\Omega^{\text{obs}}} \Pr\left(\mathbf{Y}_{nm} \mid \mathbf{Z}, \mathbf{W}; \tau\right) \prod_{k=1}^{K} \Pr\left(\mathbf{Z}_{\cdot k} \mid \beta_k, \boldsymbol{\gamma}_{\cdot k}\right) \prod_{k=1}^{K} \Pr\left(\mathbf{W}_{\cdot k}\right).
\end{aligned} \tag{12}$$

where Ω^{obs} is the collection of the indices of the observed entries of $\mathbf{Y}$. As an empirical Bayes approach, we can adaptively estimate parameters $\boldsymbol{\theta}$ by optimizing the log marginal likelihood

$$\begin{aligned}
\widehat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \ \log \Pr\left(\mathbf{Y}^{\text{obs}} \mid \boldsymbol{\theta}\right) \\
&= \arg\max_{\boldsymbol{\theta}} \ \log \int \Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\theta}\right) \, d\mathbf{Z} \, d\mathbf{W}.
\end{aligned} \tag{13}$$

Then we can infer the latent loadings and factors by obtaining their posterior probability as

$$\Pr\left(\mathbf{Z}, \mathbf{W} \mid \mathbf{Y}^{\text{obs}}; \widehat{\boldsymbol{\theta}}\right) = \frac{\Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \widehat{\boldsymbol{\theta}}\right)}{\Pr\left(\mathbf{Y}^{\text{obs}} \mid \widehat{\boldsymbol{\theta}}\right)}. \tag{14}$$

4 Fitting the Model

We first consider the single-factor case ($K = 1$). It's clear that the Bayesian inference using (13) and (14) is intractable since the marginal likelihood $\Pr\left(\mathbf{Y}^{\text{obs}} \mid \boldsymbol{\theta}\right)$ cannot be computed by marginalizing all latent variables. Here we choose variational inference (VI) [1,3] to perform approximate Bayesian inference since it's computationally efficient compared to Markov Chain Monte Carlo (MCMC)

[3]. To apply variational approximation, we first define $q(\mathbf{Z}, \mathbf{W})$ as an approximated distribution of posterior $\Pr\left(\mathbf{Z}, \mathbf{W} \mid \mathbf{Y}^{\text{obs}}; \boldsymbol{\theta}\right)$. Then we obtain the evidence lower bound (ELBO) of the logarithm of the marginal likelihood by Jensen's inequality

$$\begin{aligned}\log \Pr\left(\mathbf{Y}^{\text{obs}} \mid \boldsymbol{\theta}\right) &= \log \int \Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\theta}\right) \, d\mathbf{Z} \, d\mathbf{W} \\ &\geq \int q(\mathbf{Z}, \mathbf{W}) \log \frac{\Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\theta}\right)}{q(\mathbf{Z}, \mathbf{W})} \, d\mathbf{Z} \, d\mathbf{W} \\ &= \mathbb{E}_q\left[\log \Pr\left(\mathbf{Y}^{\text{obs}}, \mathbf{Z}, \mathbf{W} \mid \boldsymbol{\theta}\right)\right] - \mathbb{E}_q\left[\log q(\mathbf{Z}, \mathbf{W})\right] \\ &\equiv \text{ELBO}(q, \boldsymbol{\theta}),\end{aligned} \tag{15}$$

Instead of maximizing the logarithm of the marginal likelihood, we can iteratively maximize the ELBO with respect to the variational approximate posterior q and the model parameters $\boldsymbol{\theta}$

$$\left(\widehat{q}, \widehat{\boldsymbol{\theta}}\right) = \arg\max_{q, \boldsymbol{\theta}} \text{ELBO}(q, \boldsymbol{\theta}). \tag{16}$$

Using the terminology in the EM algorithm, the maximization of ELBO with respect to q is known as the E-step, and the maximization of ELBO with respect to $\boldsymbol{\theta}$ is known as the M-step. To simplify the problem, we further factorize q[1,2] as

$$q(\mathbf{Z}, \mathbf{W}) = q(\mathbf{Z}) \, q(\mathbf{W}). \tag{17}$$

Then in the E-step, without further assumptions, the optimal solution of $q(\mathbf{Z})$ and $q(\mathbf{W})$ in the E-step is given as two multivariate Gaussian distributions

$$q(\mathbf{Z}) = \mathcal{N}_N(\mathbf{Z} \mid \boldsymbol{\mu}, \mathbf{A}), \; q(\mathbf{W}) = \mathcal{N}_M(\mathbf{W} \mid \boldsymbol{\nu}, \mathbf{B}), \tag{18}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{\nu} \in \mathbb{R}^{M \times 1}$ are posterior mean vectors, $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{B} \in \mathbb{R}^{M \times M}$ are posterior covariance matrices

$$\begin{aligned}&\mathbf{A} = \text{diag}\left(\mathbf{a}^2\right), \; \boldsymbol{\mu} = \mathbf{A}\left(\beta F(\mathbf{X}) + (\boldsymbol{\tau} \circ \mathbf{Y})\, \boldsymbol{\nu}\right), \\ &\mathbf{B} = \text{diag}\left(\mathbf{b}^2\right), \; \boldsymbol{\nu} = \mathbf{B}\left(\boldsymbol{\tau} \circ \mathbf{Y}\right)^{\mathrm{T}} \boldsymbol{\mu},\end{aligned} \tag{19}$$

where $\mathbf{a}^2 \in \mathbb{R}^{N \times 1}$ and $\mathbf{b}^2 \in \mathbb{R}^{M \times 1}$ are two vectors

$$\begin{aligned}&{\mathbf{a}^2}_n = \frac{1}{\beta + \boldsymbol{\tau}_{n\cdot}^{\mathrm{T}}\left(\boldsymbol{\nu}^2 + \mathbf{b}^2\right)}, \; \boldsymbol{\mu}_n = {\mathbf{a}^2}_n \left(\beta F(\mathbf{X}_{n\cdot}) + \left(\boldsymbol{\tau}_{n\cdot} \circ \mathbf{Y}_{n\cdot}\right)^{\mathrm{T}} \boldsymbol{\nu}\right), \; n = 1, \ldots, N \\ &{\mathbf{b}^2}_m = \frac{1}{1 + \boldsymbol{\tau}_{\cdot m}^{\mathrm{T}}\left(\boldsymbol{\mu}^2 + \mathbf{a}^2\right)}, \; \boldsymbol{\nu}_m = {\mathbf{b}^2}_m \left(\boldsymbol{\tau}_{\cdot m} \circ \mathbf{Y}_{\cdot m}\right)^{\mathrm{T}} \boldsymbol{\mu}, \; m = 1, \ldots, M,\end{aligned} \tag{20}$$

and $\circ$ denotes the Hadamard product (i.e., element-wise product). In the M-step, we fix the variational approximate posterior $q(\mathbf{Z}, \mathbf{W})$ and turn to maximize the ELBO with respect to $\boldsymbol{\theta}$. We only keep the terms that are related to $\boldsymbol{\theta}$ and get

$$\begin{aligned}&\text{ELBO}(q, \boldsymbol{\theta}) \\ =&\frac{\left|\Omega^{\text{obs}}\right|}{2} \log \tau - \frac{\tau}{2}\left(\left\|\mathcal{P}_{\Omega^{\text{obs}}}\left(\mathbf{Y} - \boldsymbol{\mu}\boldsymbol{\nu}^{\mathrm{T}}\right)\right\|_{\mathrm{F}}^2 + \left\|\mathcal{P}_{\Omega^{\text{obs}}}\left(\left(\boldsymbol{\mu}^2 + \mathbf{a}^2\right)\left(\boldsymbol{\nu}^2 + \mathbf{b}^2\right)^{\mathrm{T}} - \boldsymbol{\mu}^2\left(\boldsymbol{\nu}^2\right)^{\mathrm{T}}\right)\right\|_{1,1}\right) \\ &+ \frac{N}{2} \log \beta - \frac{\beta}{2}\left(\left\|\boldsymbol{\mu} - F(\mathbf{X})\right\|_2^2 + \left\|\mathbf{a}^2\right\|_1\right) + \text{const},\end{aligned} \tag{21}$$

where $\mathcal{P}$ is a projection operator and $\mathcal{P}_\Omega(\mathbf{Y})$ outputs a matrix with the same dimension as that of $\mathbf{Y}$

$$\left(\mathcal{P}_\Omega(\mathbf{Y})\right)_{nm} = \begin{cases}\mathbf{Y}_{nm}, & \text{if } (n, m) \in \Omega, \\ 0, & \text{otherwise.}\end{cases} \tag{22}$$

By setting the derivative of the ELBO with respect to $\boldsymbol{\theta}$ as zero, we obtain the updating equations

$$\begin{aligned}\tau &= \frac{\left|\Omega^{\text{obs}}\right|}{\left\|\mathcal{P}_{\Omega^{\text{obs}}}\left(\mathbf{Y} - \boldsymbol{\mu}\boldsymbol{\nu}^{\mathrm{T}}\right)\right\|_{\mathrm{F}}^2 + \left\|\mathcal{P}_{\Omega^{\text{obs}}}\left(\left(\boldsymbol{\mu}^2 + \mathbf{a}^2\right)\left(\boldsymbol{\nu}^2 + \mathbf{b}^2\right)^{\mathrm{T}} - \boldsymbol{\mu}^2\left(\boldsymbol{\nu}^2\right)^{\mathrm{T}}\right)\right\|_{1,1}}, \\ \beta &= \frac{N}{\left\|\boldsymbol{\mu} - F(\mathbf{X})\right\|_2^2 + \left\|\mathbf{a}^2\right\|_1}.\end{aligned} \tag{23}$$

To get the linear regression coefficients, we need to solve the following least square problem

$$\gamma_0, \boldsymbol{\gamma} = \underset{\gamma_0, \boldsymbol{\gamma}}{\arg\min} \ \|\boldsymbol{\mu} - \gamma_0 - \mathbf{X}\boldsymbol{\gamma}\|_2^2 . \tag{24}$$

To extend the above algorithm to the K-factor case, we propose the following greedy algorithm which can also automatically select K

Algorithm 1: Greedy Algorithm Automatically Selecting K

Data: Main data matrix $\mathbf{Y}$ and side matrix $\mathbf{X}$
Result: Estimate of the latent loadings $\mathbf{Z}$ and factors $\mathbf{W}$

set the maximum value of the number of factors $K_{\max}$;
set the initial rank $K = 0$;
set the stop criterion sc ;
for $k = 1, \ldots, K_{\max}$ **do**
  $t \leftarrow 0$;
  repeat
    $t \leftarrow t + 1$;
    $\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)} \leftarrow \underset{\boldsymbol{\mu}, a^2; \boldsymbol{\nu}, b^2}{\arg\max}\ \mathrm{ELBO}\left(q\left(\boldsymbol{\mu}, a^2; \boldsymbol{\nu}, b^2\right); \tau^{(t-1)}, \beta^{(t-1)}; F^{(t-1)}(\cdot)\right)$;
    `// update the variational approximation of posterior`
    $\tau^{(t)}, \beta^{(t)} \leftarrow \underset{\tau, \beta}{\arg\max}\ \mathrm{ELBO}\left(q\left(\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}\right); \tau, \beta; F^{(t-1)}(\cdot)\right)$;
    `// update the model parameters`
    $\gamma_0^{(t)}, \boldsymbol{\gamma}^{(t)} \leftarrow \underset{\gamma_0, \boldsymbol{\gamma}}{\arg\max}\ \mathrm{ELBO}\left(q\left(\boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}^{(t)}, b^{2(t)}\right); \tau, \beta; F^{(t-1)}(\cdot)\right)$;
    `// update the linear regression coefficients`
  until *convergence criterion satisfied*;
  $\boldsymbol{\mu}_k, a_k^2; \boldsymbol{\nu}_k, b_k^2; \tau_k, \beta_k; \gamma_{0k}, \boldsymbol{\gamma}_{\cdot k} \leftarrow \boldsymbol{\mu}^{(t)}, a^{2(t)}; \boldsymbol{\nu}_k, b^{2(t)}; \tau^{(t)}, \beta^{(t)}; \gamma_0^{(t)}, \boldsymbol{\gamma}^{(t)}$;
  if $\mathrm{Var}\left(\boldsymbol{\mu}_k \boldsymbol{\nu}_k^{\mathrm{T}}\right) \cdot \tau_k < sc$ **then**
    break ;
  end
  $\mathbf{Y} \leftarrow \mathbf{Y} - \boldsymbol{\mu}_k \boldsymbol{\nu}_k^{\mathrm{T}}$;
  $K \leftarrow K + 1$;
end
return $\widehat{\mathbf{Z}} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K), \widehat{\mathbf{W}} = (\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_K); a_1^2, \ldots, a_K^2, b_1^2, \ldots, b_K^2$;
return $\tau_1, \ldots, \tau_K; \beta_1, \ldots, \beta_K$;
return $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K$.

5 Other methods

In order to evaluate the model we proposed, we compare it with other models in terms of prediction accuracy. In this paper, we consider other matrix factorization methods or imputation methods, like Funk SVD, Truncated SVD, SoftImpute, and KNN Imputer. And the brief introductions are as follows.

Suppose X is a large $m \times n$ matrix with many missing entries. Let Ω contains the pairs of indices (i, j) where X is observed, and let $P_\Omega(X)$ denote a matrix with the entries in Ω left alone, and all other entries set to zero. So when X has missing entries in $\Omega^\perp$, $P_\Omega(X)$ would set the missing values to zero.

5.1 Funk SVD

The Funk SVD method tries to make a matrix decomposition that can approximate the value of the real matrix as closely as possible.

$$X = AB'.$$

It solves

$$\underset{A_{n\times r}, B_{m\times r}}{\text{minimize}} \frac{1}{2} \|P_\Omega(X) - P_\Omega(AB')\|_F^2 + \frac{\lambda}{2}(\|A\|_F^2 + \|B\|_F^2)$$

5.2 SoftImpute

Consider the criterion

$$\underset{M}{\text{minimize}} \frac{1}{2} \|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda \|M\|_* ,$$

where $\|M\|_*$ is the nuclear norm of $\mathcal{M}$ (sum of singular values).

If $\widehat{M}$ solves this convex-cone problem, then it satisfies the following stationarity condition:

$$\widehat{M} = S_\lambda(Z)$$

where

$$Z = P_\Omega(X) + P_{\Omega^\perp}(\widehat{M}).$$

Hence Z is the "filled-in" version of X. The operator $S_\lambda(Z)$ applied to matrix Z does the following:

1. Compute the SVD of $Z = UDV'$, and let d_i be the singular value in D.
2. Soft-threshold the singular values: $d_i^* = (d_i - \lambda)_+$.
3. Reconstruct: $S_\lambda(Z) = UD^*V'$. We call this operation the "soft-thresholded SVD". Notice that for sufficiently large λ,D^* will be rank-reduced, and hence so will be UD^*V'.

5.3 Truncated SVD

Suppose our matrix X ($X = P_\Omega(X)$) has the singular value decomposition

$$X = U\Sigma V'.$$

This can be written as

$$X = \sum_{i=1}^{p} \sigma_i u_i v_i',$$

where the σ_i are the singular values and p is the number of singular values that are non-zero. The u_i and v_i are the i-th columns of U and V respectively. Singular values are nonnegative, and listed in decreasing order.

Chosen k such that $k < rank(X)$, then we can construct the so-called truncated SVD,

$$X_k = \sum_{i=1}^{k} \sigma_i u_i v_i'.$$

5.4 KNN Imputer

KNN Imputer performs nearest neighbor imputations which weights samples using the mean squared difference on features for which two rows both have observed data.

6 Results

6.1 Impute accuracy

First, we split the non-missing ratings of users' rating matrix into two parts: training data and testing data, for 50 perturbations. Then we apply five methods (Proposed Matrix Factorization, Funk SVD,

Soft Impute, KNN Impute and Truncated SVD) to fit in the training data and compare their testing MAEs (figure 1), and we can find that our model achieves higher prediction accuracy compared to other methods with the utilization of movie genre information.

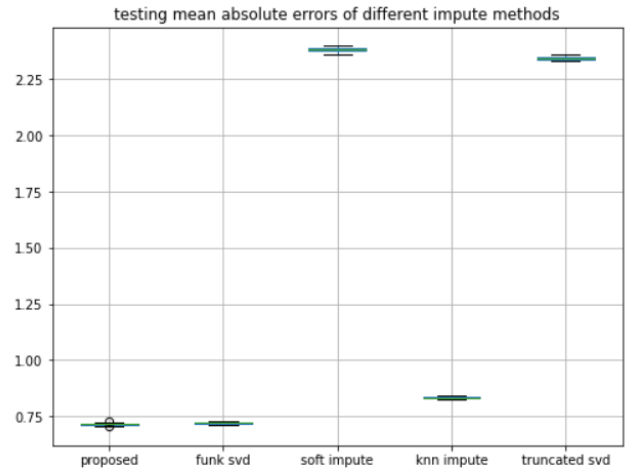

Figure 1: The boxplot of mean absolute errors in different methods

6.2 Linear regression of factors

In this report, we explore how the scores of movies on latent factors relate to the genres of movies by performing statistical analyses of linear regression. We get matrix Z from Proposed Matrix Factorization, then we perform the linear regression of first three factors $Z_{.k}(k = 1, 2, 3)$ on X(e.g. the movies information), see (11). The following figures are the bar plots of linear regression coefficients (removing the intercept and unknown). And table 1 is the p-values of linear regression coefficients and multiple R squared.

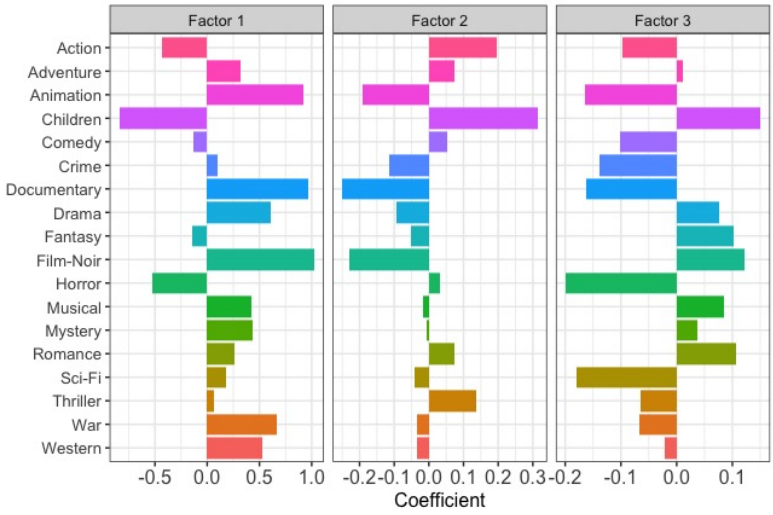

Figure 2: The barplots of regression coefficients for the top three factors.

Table 1: Proposed Matrix Factorization

	factor 1	factor 2	factor 3
Action	3.30E-09	< 2E-16	< 2E-16
Adventure	0.000772	0.000255	0.351925
Animation	3.54E-08	9.21E-08	2.62E-14
Children	6.58E-15	< 2E-16	< 2E-16
Comedy	0.027899	4.11E-05	< 2E-16
Crime	0.2729	7.29E-09	< 2E-16
Documentary	8.16E-12	< 2E-16	< 2E-16
Drama	< 2E-16	2.41E-13	< 2E-16
Fantasy	0.488231	0.234161	0.000127
Film-Noir	1.78E-07	4.49E-08	1.48E-06
Horror	6.46E-07	0.159209	< 2E-16
Musical	0.001457	0.563702	5.06E-07
Mystery	0.000431	0.851731	0.018383
Romance	5.70E-05	1.22E-07	< 2E-16
Sci-Fi	0.074736	0.061959	< 2E-16
Thriller	0.382129	< 2E-16	2.44E-12
War	3.32E-09	0.152848	2.40E-06
Western	0.003097	0.371944	0.330681
Multiple R-squared	0.249	0.3975	0.5762

As we can see, the p-values of most regression coefficients are significantly, which means factors $Z_{\cdot k}$ are highly correlated to the movie genres. For factor 1, the coefficients of "Animation", "Children", "Documentary", "Drama", "Film-Noir", "Horror", "War" and "Western" are relatively large, which represents that these genres play important roles in factor 1. For factor 2, "Action", "Animation", "Children", "Documentary", "Film-Noir", and "Thriller" are relatively large, which represents that these genres play important roles in factor 2. For factor 3, "Animation", "Children", "Crime", "Documentary", "Film-Noir", "Horror", and "Sci-Fi" are relatively large, which represents that these genres play important roles in factor 3. In addition, the results indicate that our model can well cope with the movie-level cold-start problem. Considering a new movie put out, we can get the genres of the new movie but have no information about users' ratings. Since the movie's latent factors are highly correlated to the genres of the movie in our model, we can apply regression functions to estimate the new movie's latent factors and further estimate the users' ratings on the movie. Then, we can recommend the movie to the existing users who have top estimated users' ratings.

6.3 Explore embeddings in user preference factorization matrix

In addition, we view the decomposed matrix W represented preferences of users on latent factors as a kind of embedding of users, and we conduct dimension reduction methods to find the lower dimensional embedding for users' preferences on movies and further explore its relationship with users' features by visualization. At first, we try to classify the age and occupation into 5 and 4 categories, respectively. Then we choose the top 2 factors to plot the data and also use the MDS, Isomap, modified LLE, Laplacian Eigenmap, and tSNE to perform data reduction and visualization. According to the results, we find that the users' preferences on movies seem to have no obvious relationship with gender, age, and occupation. Here, we only give out the visualization with the top 2 factors with respect to occupation, see Figure 5.

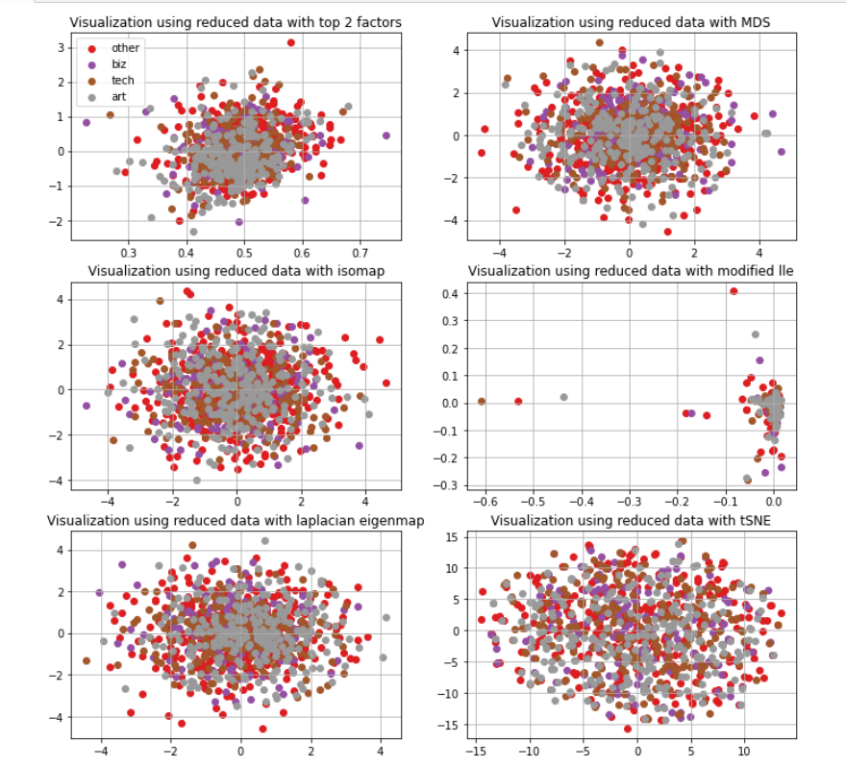

Figure 3: Visualization with respective to occupation

7 Summary

In this report, we do probabilistic matrix factorization with the incorporation of movie genre information on MovieLens 100K Dataset. Our model achieves higher prediction accuracy compared to other methods with the utilization of movie genre information since it allows the model parameters to be automatically estimated under the empirical Bayes framework. Also, our model is computationally efficient and scalable to large datasets by exploiting variational inference. In addition, our model can well cope with the movie-level cold-start problem to some extent.

8 Contribution

All the group members discussed and shared ideas throughout the whole project. WANG Zhiwei raised the scientific problems, proposed the model, derived the algorithm, implemented the methods, designed the research plan, and finished PPT. QIU Zhenyu wrote the code, did the experiments, compared different methods, and visualized and analyzed the results. CAI Bibi (Results, Other methods), QIU Zhenyu (Abstract, Introduction), and WANG Zhiwei (Model, Methods, PPT) wrote the report together.

References

[1] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[2] Blei, D. M., A. Kucukelbir & J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association* **112**(518), 859–877.

[3] Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods.* Department of Computer Science, University of Toronto, Toronto, ON, Canada.

[4] Mnih, A. & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. *Advances in neural information processing systems, 20*.