Bayesian Probabilistic Matrix Factorization For Recommendation Systems

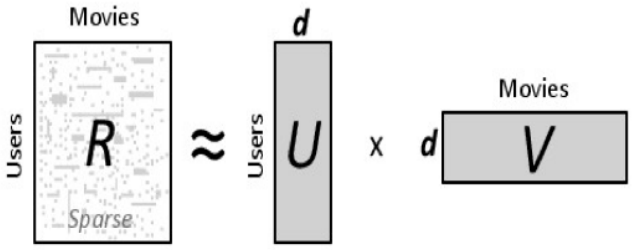
Yatish Goel
Department of Civil Engineering
IIT KANPUR
yatishg@iitk.ac.in

1 ABSTRACT

These days we are constantly being recommended from variety of sources, such as what blog to read, what music to listen to etc. And these recommendation systems are becoming more personalized than ever. In the era of digital world, we see recommendation in every area whether it is e-commerce website or entertainment website or social network site. Recommendation not only gives a user his recommended choice (based on past activity) but it also tells about user behaviour. We use matrix factorization technique in recommendation systems. In this matrix factorization technique a sophisticated recommender system is built which outperforms nearest-neighbour techniques (In the setting of movie recommendation system). There are two types of approaches which are used in recommendation system:
a) Content-based filtering (it makes recommendations based on item content)
b) Collaborative filtering (it does not make recommendations based on item content)
Low-rank matrix approximation methods provide one of the simplest and most effective approaches to Collaborative filtering. Such models are usually fitted to data by finding an MAP estimate of the model parameters, a procedure that can be performed efficiently even on very large datasets. However, unless the regularization parameters are tuned carefully, this approach is prone to overfitting. In this paper, we present a full Bayesian treatment of the Probabilistic Matrix Factorization (PMF) model in which model capacity is controlled automatically by integrating over all model parameters and hyper parameters.

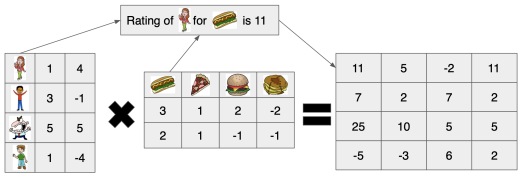
2 INTRODUCTION

Matrix factorization (MF) techniques are commonly applied to model sparse data matrices. The low-rank assumption of MF methods decomposes the data matrix as the product of two lower rank matrices. By construction, each entry in the data matrix is the inner product of a vector from each of the two lower rank matrices. This has practical implementation in recommender systems where rows and columns corresponds to the set of objects, e.g.: Netflix, Facebook and Genetics also.
It is basically used for calculation of complex matrix operations. Division of matrix is done such that if we multiply factorized matrices, we will get original matrix. It is used for discovering latent features between two entities (It can be used for more than two entities but it will come under tensor factorization). A user's rating of an item is modelled by the inner product of an item factor vector and a user factor vector. This means that the N $\times$ M preference matrix of ratings that N users assign to M movies is modeled by the product of an D $\times$ N user coefficient matrix U and a D$\times$M factor matrix V. Training such a model amounts to finding the best rank-D approximation to the observed N $\times$ M target matrix R under the given loss function.



Decomposition of matrix R in two low Rank matrices U and V

In these models, factor variables are assumed to be marginally independent while rating variables are assumed to be conditionally independent given the factor variables. The main drawback of such models is that inferring the posterior distribution over the factors given the ratings is intractable. MAP estimate can be computed of the model parameters by maximizing log-posterior over model parameters. Here we are interested in predicting for new user/item pair rather model parameters. This view suggests taking a Bayesian approach to the problem which involves integrating out the model parameters.
Consider the setting where we have a data matrix **D** with N rows and M columns where R has missing values throughout.
One way to formalize this task is the matrix completion problem where we try to replace the missing data with knowledge of the known values. A popular model based approach is to assume that the data matrix R has low rank and hence can be factorized into the product of two low rank matrices U, V.



Product of two low rank matrices

3 Probabilistic Matrix Factorization

Probabilistic Matrix Factorization (PMF) is a probabilistic linear model with Gaussian observation noise
Suppose we have N users and M movies. Let R_{ij} be the rating value of user i for movie j, U_i and V_j represent D-dimensional userspecific and moviespecific latent feature vectors respectively. The conditional distribution over the observed ratings $R \in \mathbb{R}^{NXM}$ (the likelihood term) and the prior distributions over $U \in \mathbb{R}^{DXN}$ and $V \in \mathbb{R}^{DXM}$ are given by:

$$p(R|U,V,\alpha) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1})\right]^{I_{ij}}$$

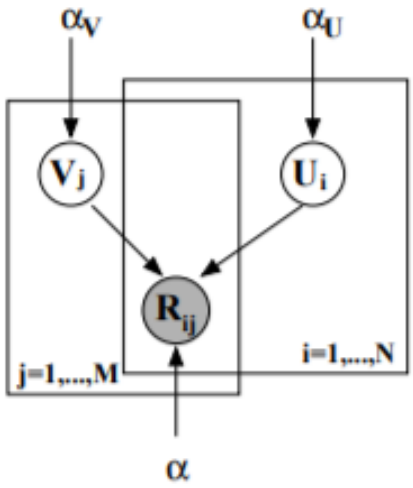
Here we assume elements are independent, and they share common variance. And now we define priors distribution on U and V

$$p(U|\alpha_U) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\alpha_U^{-1}I)$$

$$p(V|\alpha_V) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\alpha_V^{-1}I)$$

Where $\mathcal{N}(x|\mu,\alpha^{-1})$ denotes the Gaussian distribution with mean μ and precision α and I_{ij} is the indicator variable that is equal to 1 if user i rated movie j and equal to 0 otherwise

4 Posterior Distribution



Graphical Model of PMF

$$p(U,V|R,\alpha,\alpha_u,\alpha_v) \\ \alpha\ \mathcal{N}(R|U,V,\alpha)p(U|\alpha_u)p(V|\alpha_v) \quad (1)$$

Now we can find matrices U and V that maximise this posterior diisribution. We can maximize posterior by maximizing log posterior

$$\begin{aligned}&\log\ p(U,V|R,\alpha,\alpha_u,\alpha_v)\ \ \alpha \\ &-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-u_i^T v_j)^T\alpha(R_{ij}-u_i^T v_j) \\ &\frac{-\alpha_u}{2}\sum_{i=1}^{N} u_i^T u_i\ \ -\frac{\alpha_u}{2}\sum_{i=j}^{M} v_j^T v_j\end{aligned} \tag{2}$$

To estimate U and V we can use gradiant decent. Now differentiating log posterior wrt u_i and v_j

$$\frac{\partial}{\partial u_i}\mathcal{L}(U,V)\ =\ \alpha\sum_{j=1}^{M} I_{ij}(R_{ij}-u_i^T v_j)v_i\ +\ \alpha_U u_i \tag{3}$$

$$\frac{\partial}{\partial v_j}\mathcal{L}(U,V)\ =\ \alpha\sum_{i=1}^{N} I_{ij}(R_{ij}-u_i^T v_j)u_i\ +\ \alpha_v v_j \tag{4}$$

Now iteratively update until it converges

First input data matrix D, learning rate η, and initial estimates $U^{(0)}$ and $V^{(0)}$

for t= 0, 1....., do

for i = 1, 2, 3...., n do

$\mathbf{u_i^{(t+1)}} = \mathbf{u_i^{(t)}} - \eta\frac{\partial}{\partial \mathbf{u_i}}\mathcal{L}(\mathbf{U},\mathbf{V})$

end for

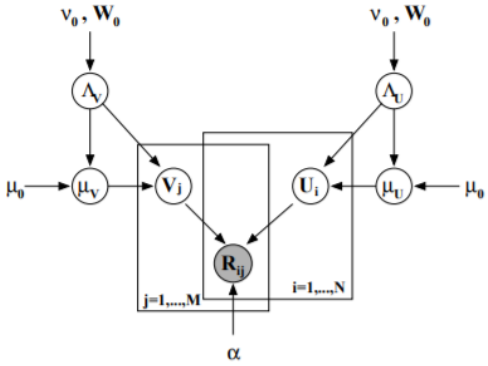
for j = 1, 2 ....., p do

$\mathbf{v_j^{(t+1)}} = \mathbf{v_j^{(t)}} - \eta\frac{\partial}{\partial \mathbf{v_j}}\mathcal{L}(\mathbf{U},\mathbf{V})$

end for

end for

The problem here is of model complexity. To solve this problem we could introduce priors on hyper paranmeters and maximize the log posterior of the model over both parameters and hyperparameters which allows model complexity to be controlled automatically based on training data



Graphical Model of Bayesian Probabilistic Matrix Factorization

The prior distribution over user and movie feature vectors and assumed to be gaussian

$$p(U|\mu_u,\Lambda_u)\ =\ \prod_{i=1}^{N}\mathcal{N}(U_i|\mu_U,\Lambda_U^{-1}) \tag{5}$$

$$p(V|\mu_v,\Lambda_v)\ =\ \prod_{j=1}^{M}\mathcal{N}(V_j|\mu_V,\Lambda_V^{-1}) \tag{6}$$

We assume Gaussian-Wishart priors on user and movie hyperparameter $\Theta_U\ =\ \{\mu_U,\Lambda_U\}$ And $\Theta_V\ =\ \{\mu_V,\Lambda_V\}$

$$\begin{aligned}p(\Theta_U|\Theta_0)\ &=\ p(\mu_U|\Lambda_U)p(\Lambda_U) \\ &=\mathcal{N}(\mu_0,(\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0,\nu_0)\end{aligned} \tag{7}$$

$$\begin{aligned}p(\Theta_V|\Theta_0)\ &=\ p(\mu_V|\Lambda_V)p(\Lambda_V) \\ &=\mathcal{N}(\mu_0,(\beta_0\Lambda_V)^{-1})\mathcal{W}(\Lambda_V|W_0,\nu_0)\end{aligned} \tag{8}$$

$$\begin{aligned}&\mathcal{W}(\Lambda|W_0,\nu_0) \\ &=\ \frac{1}{C}|\Lambda|^{\frac{\nu_0-D-1}{2}}\exp\left(-\frac{1}{2}Tr(W_0^{-1}\Lambda)\right)\end{aligned} \tag{9}$$

We set $\nu_0 = D$ and W_0 to the identity matrix for both user and movie hyper parameters and choose $\mu_0 = 0$ by symmetry. Here $\Theta_0\ =\ \{\mu_0,\nu_0,W_0\}$

5 PREDICTION

The prediction on the rating value R^*_{ij} for user i and movie j is obtained by marginalizing over model parameters and hyperparamters:

$$\begin{aligned} &p(R^*_{ij}|R,\Theta_0) \\ &= \iint p(R^*_{ij}|U_i,V_j)p(U,V|R,\Theta_U,\Theta_V) \\ &p(\Theta_U,\Theta_V|\Theta_0)d\{U,V\}d\{\Theta_U,\Theta_V\} \end{aligned} \tag{10}$$

It can be seen that it is intractable due to complexity of posterior so we need to resort to approximate inference.

We can use monte carlo approximation to the predictive distribution

$$p(R^*_{ij}|R,\Theta_0) \approx \frac{1}{K}\sum_{i=1}^{K} p(R^*_{ij}|U_i^{(k)},V_j^{(k)}) \tag{11}$$

The samples $\{U_i^{(k)}, V_j^{(k)}\}$ are generated by running a Markov chain whose stationary distribution is the posterior distribution over the model parameters and hyperparameters $\{U, V, \Theta_U, \Theta_V\}$. The advantage of the Monte Carlo-based methods is that asymptotically they produce exact results.

6 INFERENCE

For inference we need to integrate out the features, the hyper parameters. The integral is computationally intractable and it requires approximation methods. We we will use MCMC algorithms in gibbs sampling algorithm. Use of non conjugate distribution can be computationally expensive so we use conjugate prior distribution that yields tractable posterior distribution that are easy to sample form

$$\begin{aligned} &p(U_i^*|R,\Theta_U,\alpha) \\ &= \prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^TV_j,\alpha^{-1})\right]^{I_{ij}} p(U_i|\mu_U,\Lambda_U) \\ &= \prod_{j=1}^{M}\exp\Big\{-\frac{\alpha}{2}\Big[(r_{ij}^Tr_{ij}) - (2r_{ij}V_j^TU_i + (U_i^TV_jV_j^TU_i)\Big] \\ &-\frac{\Lambda_u}{2}(U_i^TU_i - \mu_u^TU_i - U_i^T\mu_m + \mu_u^T\mu_u\Big\} \end{aligned} \tag{12}$$

Ignoring terms independent of U_i

$$\begin{aligned} &p(U_i^*|R,\Theta_U,\alpha) \\ &= \prod_{j=1}^{M}\exp\Big\{\alpha r_{ij}V_j^TU_i - \frac{\alpha}{2}(U_i^TV_jV_j^TU_i) \\ &-\frac{\Lambda_u}{2}(U_i^TU_i) + \frac{\Lambda_u}{2}(\mu_u^TU_i) + \frac{\Lambda_u}{2}(U_i^T\mu_u) - \frac{\Lambda_u}{2}(\mu_u^T\mu_u)\Big\} \end{aligned} \tag{13}$$

By comparing with $(U_i-\mu_i^*)^T\Lambda^*(U_i-\mu_i^*)$

$$p(U_i|R,V,\Theta_U,\alpha) = \mathcal{N}(U_i|\mu_i^*,[\Lambda_i^*]^{-1}) \tag{14}$$

$$\begin{aligned} \Lambda_i^* &= \Lambda_u + \alpha\sum_{j=1}^{M}[V_jV_j^T]^{I_{ij}} \\ \mu_i^* &= [\Lambda_i^*]^{-1}(\alpha\sum_{j=1}^{M}[V_jR_{ij}]^{I_{ij}} + \Lambda_u\mu_u) \end{aligned} \tag{15}$$

So

$$p(U|R,V,\Theta_U) = \prod_{i=i}^{N} p(U_i|R,V,\Theta_U) \tag{16}$$

The conditional distribution over the user hyper parameters conditioned on the user feature matrix U is given by the Gaussian-Wishart distribution

$$\begin{aligned} &p(\mu_U,\Lambda_U|U,\Theta_0) \\ &\quad \alpha\prod_{i=1}^{N}p(U_i|\mu_U,\Lambda_U) \\ &\quad p(\mu_U|\mu_0,\beta_0\Lambda_0)p(\Lambda_U|\nu_0,W_0) \\ &logp(\mu_U,\Lambda_U|U,\Theta_0) = \frac{N}{2}log|\Lambda_U| \\ &\quad -\frac{1}{2}\sum_{i=1}^{N}(U_i-\mu_U)^T\Lambda_U(U_i-\mu_U) \\ &\quad +\frac{1}{2}log|\Lambda_U| \\ &\quad -\frac{\beta_0}{2}(\mu_U-\mu_0)^T\Lambda_U(\mu_U-\mu_0) \\ &\quad -\frac{1}{2}tr(W_0^{-1}\Lambda_U) + \frac{\nu_0-d-1}{2}log\ |\Lambda_U| \end{aligned} \tag{17}$$

So

$$\begin{aligned} &p(\mu_U,\Lambda_U|U,\Theta_0) = \\ &\quad \mathcal{N}(\mu_U|\mu_0^*,(\beta_0^*\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0^*,\nu_0^*) \end{aligned} \tag{18}$$

where

$$\mu_0^* = \frac{\beta_0\mu_0 + \bar{U}N}{\beta_0 + N} \ , \ \beta_0^* = \beta_0 + N \ , \ \nu_0^* = \nu_0 + N$$
$$[W_0^*]^{-1} = W_0^{-1} - N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}(\mu_0 - \bar{U})(\mu_0 - \bar{U})^T$$
$$\bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i \ , \ \bar{S} = \frac{1}{N}\sum_{i=1}^{N} U_i U_i^T \tag{19}$$

Gibbs sampling for Bayesian PMF

1. Initialize model parameters $\{U^1, V^1\}$
2. For t=1,...,T
   - Sample the hyperparameters (Eq. 18):
   $$\Theta_U^t \sim p(\Theta_U | U^t, \Theta_0)$$
   $$\Theta_V^t \sim p(\Theta_V | V^t, \Theta_0)$$
   - For each $i = 1, ..., N$ sample user features in parallel (Eq. 14):
   $$U_i^{t+1} \sim p(U_i | R, V^t, \Theta_U^t)$$
   - For each $i = 1, ..., M$ sample movie features in parallel:
   $$V_i^{t+1} \sim p(V_i | R, U^{t+1}, \Theta_V^t)$$

7 EXPERIMENT

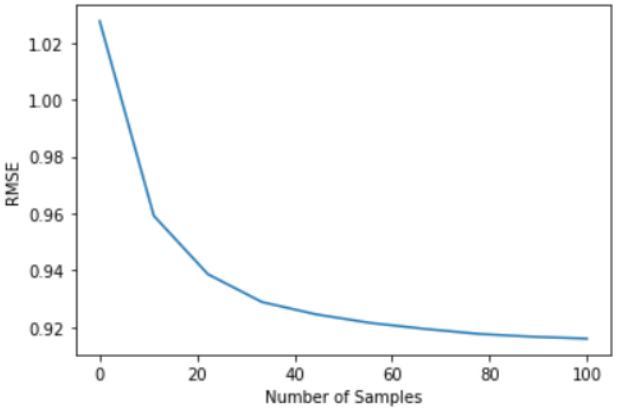
The dataset used here represent the distribution of all ratings Netflix. This data contains 10000 ratings from 943 users on 1682 movies. Each user has rated at least 20 movies. Performance is assessed by submitting predicted ratings to Netflix which then posts the root mean squared error (RMSE) on an unknown half of the test set. As a baseline, Netflix provided the test score of its own system trained on the same data, which is 0.9514

7.1 Training BPMF Model

We initialized the Gibbs sampler by setting the model parameters U and V to their MAP estimates obtained by training a linear PMF model. We also set $\mu_0 = 0$, $\nu_0 = D$, *and* W_0 to o the identity matrix, for both user and movie hyper priors. The observation noise precision α was fixed at 2. The predictive distribution was computed using Eq. 11 by running the Gibbs sampler with samples $\{U_i^{(k)}, V_j^{(k)}\}$ collected after full Gibbs step.

7.2 Results

We used 50-D feature vectors and obtained a RMSE of 0.9162



7.3 Conclusion

We have presented a fully Bayesian treatment of Probabilistic Matrix Factorization by placing hyperpriors over the hyperparameters and using MCMC methods to perform approximate inference.Bayesian model is that it provides a predictive distribution instead of just a single number, allowing the confidence in the prediction to be quantified and taken into account when making recommendations using the model

8 REFERENCES

Ruslan Salakhutdinov, Andriy Mnih . Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo