Flexible Collaborative Filtering: A Bayesian Approach

DIMAS SOARES LIMA, Universidade Federal do Rio de Janeiro, Brasil
MARINA SILVA PAEZ, Universidade Federal do Rio de Janeiro, Brasil
HUGO TREMONTE DE CARVALHO, Universidade Federal do Rio de Janeiro, Brasil

Recommendation systems seek to predict the rating or preference that a user would give to an item. We propose a new method for collaborative filtering that allows flexible recommendations to users through Markov Chain Monte Carlo algorithms (MCMC). With this approach, one can draw samples from the predictive posterior distribution and use it to produce point estimates, since after convergence each sampled value can be used as rate prediction. Our proposal allows fast results to be displayed since it does not require waiting for the simulation of a full chain before making predictions. This is not only welcome by the users themselves but also helps with the learning mechanism of the algorithm. Also, one of the biggest concerns of this study was to create an algorithm that is scalable. To do so, we propose a Bayesian optimization step within the MCMC algorithm, in order to circumvent a costly matrix inversion. Finally, an application to the Movie Lens data set [7] is presented as an illustration, and results comparable to the state-of-the-art were obtained.

CCS Concepts: • **Mathematics of computing → Probability and statistics**; **Bayesian computation**.

Additional Key Words and Phrases: Recommender System, Markov Chain Monte Carlo, Bayesian Optimization

ACM Reference Format:
Dimas Soares Lima, Marina Silva Paez, and Hugo Tremonte de Carvalho. 2020. Flexible Collaborative Filtering: A Bayesian Approach. In *The ACM Conference Series on Recommender Systems, Sept 22–26, 2020, Online*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Custom recommendations are often obtained from lists of ranked items. When building a ranking, recommendation systems try to predict which product or service most likely fits the user preferences based on their background. To this end, the system must constantly collect information from users and update their preferences dynamically. This information can be explicit (such as grades assigned to a product), or inferred from user actions (time spent on a webpage, number of clicks, recent searches). In both cases, the system learns based on users' feedback: an option is recommended for an user and the system automatically learns through his positive or negative reaction to it.

Several model-based approaches have been proposed through matrix factorization methods within the context of collaborative filtering ([2], [16]), but in most of them the system is limited to a fixed ranking per user and suffers from the well known cold start problem. Some approaches deal with the latter problem in an interactive way (see [9] and [15]).

Dimension reduction is one of the tools used in the recommendation system scheme. Often, a recommendation problem can contain high-dimensionality structures. Several sophisticated methods were proposed and discussed in

Manuscript submitted to ACM

[16] (Alternating least squares - ALS), [2], [11] and [1]. These models leverage well-known dimensionality reduction methods to fill in the missing entries. An elegant application of ALS in the context of Latent Factor Models was proposed by [16]. In this approach, the goal is to find a vector $\mathbf{x}_i \in \mathbb{R}^f$ for each user i and a vector $\mathbf{y}_j \in \mathbb{R}^f$ for each item j that will factor user preferences. In other words, preferences are assumed to be the inner products $r_{ij} = \mathbf{x}_i^T \mathbf{y}_j$. Vectors $\mathbf{x}_i$ and $\mathbf{y}_j$ are known as the user-factors and item-factors, respectively.

Another approach was presented in [14], where linear models were used in the recommender system scenario. The authors compare their method to another model-based method using decision trees, and to memory-based methods using data from several domains. In this paper we propose a new collaborative filtering approach for recommendation systems. We focus on a model-based recommendation approach similar to the one proposed by [14], and with the same core algorithm, but under the Bayesian paradigm. In the application section, we compare our algorithm to the dimensionality reduction-based algorithm ALS, and to the Probabilistic Matrix Factorization - PMF [10].

Our model assumes that the ratings matrix, with n rows (total number of users) and p columns (total number of items) follows a matrix variate normal distribution. Note that in general we only observe only a few entries of this matrix, and our challenge is to predict the missing scores in order to make the recommendations. To do so, we use a conditional construction that specifies a linear regression to each column of the ratings matrix, assuming that part of the variability of the ratings given to a fixed item is explained by the ratings given to the other items. Also, one can penalize the parameters in the regression structure depending on the goal of the recommendation system of interest. Our final purpose is to rank the items based on the predictive distribution of the missing ratings, and make the recommendations following these results. This approach has the advantage of allowing new recommendations even if the user has not interacted with the system yet. We propose Markov Chain Monte Carlo (MCMC, [4]) methods to sample from the posterior and the predictive distributions. In this scenario, it is possible to use a fully Bayesian approach to sample from the full posterior distribution. A recurrent problem in this approach is the intense computational demand. We propose a semi-Bayesian approach using Bayesian optimization to obtain a point estimate of a set of parameters in the algorithm structure at each iteration as opposed to drawing from it, aiming to improve the computational time processing the algorithm.

Under this approach, after convergence, the algorithm can provide different and varied recommendations to the user, which can be desirable. For example, the system can give a new recommendation to the same user when he (or she) updates the website, instead of always providing the same one. Indeed, under the Bayesian paradigm, any kind of flexible prediction can be used.

2 PROPOSED MODEL

In this section we present the proposed model and it's interpretation. To build the model we assume that the ratings given by different users to a fixed item are conditionally independent given all other ratings. However, ratings given by the same user to different items are correlated. This is a reasonable assumption as a user will have a particular preference and will tend to rate similar items in a similar way. We also assume that the ratings can be seen as approximately Normal, with every item assuming a different mean and variance. Therefore the matrix of ratings will follow a Normal matrix variate distribution (see [6]).

Our method takes advantage of the fact that, after some algebra, the vector of ratings given to each item can be written as a regression of the ratings given to the other items. This result will be developed below. These regression structures are explored inside the proposed MCMC method. Note, however, that our algorithm must be able to deal

with both high dimension (large number of users) and several covariates (large number of items), which are usual in recommendation systems.

Similar approaches to ours can be found in the literature under the classic paradigm, such as [13] and [14]. We, on the other hand, follow a Bayesian approach that makes the structure more flexible, and allows to use each sample from the posterior as a recommendation structure.

Suppose n users rated p items, and let $y_{i,j}$ be the rating given by user i to item j. Denote by $(y_{i,1}, y_{i,2}, \cdots, y_{i,p})^T$ the vector of ratings provided by the i-th user, with $i = 1, \ldots, n$. As mentioned before, we assume these ratings are normally distributed, with the ratings of item j having mean μ_j and variance ϕ_j^2, $j = 1, \ldots, p$. We also assume that the ratings given by different users to an item are independent conditionally on $\boldsymbol{\mu}$, Φ and other item' ratings, that is

$$(y_{i,1}, y_{i,2}, \cdots, y_{i,p})^T \overset{\text{iid}}{\sim} N_p(\boldsymbol{\mu}, \Phi), \qquad i = 1, \ldots, n,$$

with $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_p)^T$ and

$$\Phi = \begin{pmatrix} \phi_1^2 & \phi_{1,2} & \cdots & \phi_{1,p} \\ \phi_{2,1} & \phi_2^2 & \cdots & \phi_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p,1} & \phi_{p,2} & \cdots & \phi_p^2 \end{pmatrix}, \tag{1}$$

where $N_p(\boldsymbol{\mu}, \Phi)$ denotes a p-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Φ.

Now, denote by Y_k the vector of ratings given by all the users to the k-th item, such that $Y_k = (y_{1,k}, y_{2,k}, \ldots, y_{n,k})^T$, $k = 1, 2, \ldots p$, and denote by $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_p) \in \mathbb{R}^{n \times p}$ the matrix of ratings provided by the n users to the p item. Therefore,

$$\mathbf{Y} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,p} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & \cdots & y_{n,p} \end{pmatrix} \tag{2}$$

is a random matrix with mean matrix M and covariance matrix $\Sigma = \Phi \otimes I_n$, that is, when written as a vector

$$vec(\mathbf{Y}) = \begin{pmatrix} Y_1^T & Y_2^T & \ldots & Y_p^T \end{pmatrix}^T = \begin{pmatrix} y_{1,1} & y_{2,1} & \ldots & y_{n,1} & y_{1,2} & y_{2,2} & \ldots & y_{n-1,p} & y_{n,p} \end{pmatrix}^T \sim N_{np}(\boldsymbol{\mu}, \Sigma) \tag{3}$$

with

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 & \mu_1 & \ldots & \mu_1 & \mu_2 & \mu_2 & \ldots & \mu_p & \mu_p \end{pmatrix}^T, \tag{4}$$

where I_n denotes the n-dimensional identity matrix and $\otimes$ represents the Kronecker product. More details about matrix variate distributions can be found in [6]. Here $\boldsymbol{\mu} = vec(M)$. The covariance matrix Σ of the vectorized matrix $vec(\mathbf{Y})$ is given by

$$\Sigma = \Phi \otimes I_n = \begin{pmatrix} \phi_1^2 I_n & \phi_{1,2} I_n & \cdots & \phi_{1,p} I_n \\ \phi_{1,2} I_n & \phi_2^2 I_n & \cdots & \phi_{2,p} I_n \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{1,p} I_n & \phi_{1,p-1} I_n & \cdots & \phi_p^2 I_n \end{pmatrix}. \tag{5}$$

Defining

$$Y_{-k} = vec(Y_1, \ldots Y_{k-1}, Y_{k+1}, \ldots, Y_p)^T, \tag{6}$$

a vector with mean $\boldsymbol{\mu}_{-k}$ and covariance matrix $\Sigma_{-k,-k}$, we have that ([8])

$$\begin{pmatrix} Y_k \\ Y_{-k} \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_k \\ \boldsymbol{\mu}_{-k} \end{pmatrix}, \begin{pmatrix} \Sigma_{k,k} & \Sigma_{k,-k} \\ \Sigma_{-k,k} & \Sigma_{-k,-k} \end{pmatrix}\right) \tag{7}$$

and

$$Y_k|(Y_{-k} = y_{-k}) \sim N_n(\boldsymbol{\mu}^k, \Sigma^k) \tag{8}$$

with

$$\boldsymbol{\mu}^k = \boldsymbol{\mu}_k + \Sigma_{k,-k}\Sigma^{-1}_{-k,-k}(y_{-k} - \boldsymbol{\mu}_{-k}), \tag{9}$$

$$\Sigma^k = \Sigma_{k,k} - \Sigma_{k,-k}\Sigma^{-1}_{-k,-k}\Sigma_{-k,k}. \tag{10}$$

Note that the model structure allows us to describe each column of the ratings matrix as a function of the other columns. Therefore, it is possible to reparametrize the distribution above in the following way

$$Y_k|Y_{-k} \sim N_n(\mathbf{1}\beta_1^{(k)} + X^{(k)}\boldsymbol{\beta}^{(k)}, \sigma_k^2 I_n), \tag{11}$$

where $\mathbf{1}$ is a n-dimensional vector of ones; $\beta_1^{(k)}$ is a scalar; $\boldsymbol{\beta}^{(k)}$ is a vector of coefficients with dimension $p-1$; $X^{(k)}$ is the ratings matrix $\mathbf{Y}$ without the column k; and σ_k^2 is defined as the variance of $Y_{ik}|Y_{-k}$.

The model when written as above has the form of a multiple regression. Instead of assigning prior distributions to the parameters under the matrix variate formulation of the model, we prefer to work under this more convenient construction. Therefore, to complete the model specification, we assign independent prior distributions for the parameters $\beta_1^{(k)}$, $\boldsymbol{\beta}^{(k)}$ and σ_k^2, $\forall k$. As we do not want to assume any prior knowledge about some parameters, we assign a flat distribution as a prior for the intercept $\beta_1^{(k)}$ and a non informative Jeffreys' prior for σ_k^2. Also, given the large number of covariates in each regression and the fact that almost always the k-th item will have only a few users who interacted with it, we propose a Laplace prior distribution for the regression coefficients $\boldsymbol{\beta}^{(k)}$. Also, [12] reports that with this structure the maximum of the posterior distribution is the same as obtained by the LASSO penalization. With this prior, we are penalizing the parameters and dealing with the $p > n$ problem.

The proposed model can be written as:

$$Y_k|\, Y_{-k}, \beta_1^{(k)}, \boldsymbol{\beta}^{(k)}, \sigma_k^2 \sim N_{n_k}(\boldsymbol{\beta}_1^{(k)} + X^{(k)}\boldsymbol{\beta}^{(k)}, \Sigma^k) \qquad k = 1, \ldots, p, \tag{12}$$

$$\beta_1^{(k)}|\,. \sim flat \qquad k = 1, \ldots, p, \tag{13}$$

$$\beta_j^{(k)}|\,\sigma_k^2, \lambda_k \sim La\left(0, \frac{2\sigma_k^2}{\lambda_k}\right), \qquad j = 1, \ldots, p-1, k = 1, \ldots, p, \tag{14}$$

$$p(\sigma_k^2) \propto \frac{1}{\sigma_k^2}, \qquad k = 1, \ldots, p, \tag{15}$$

$$\lambda_k \sim G(\alpha_0, \gamma_0), \qquad k = 1, \ldots, p, \tag{16}$$

where n_k denotes the number of users that have rated the k^{th} item. It can be shown that the joint distribution of $\mathbf{Y}$ exists with this construction (see [3]), but the posterior distribution does not have a known form. That way, we propose a MCMC algorithm to sample from the model parameters. To do so, it is useful to obtain the full conditional distributions,

which are given below:

$$
\begin{aligned}
p(\boldsymbol{\beta}^{(k)}|\mathbf{Y}, \lambda_k, \sigma_k^2) &\propto \exp\left\{ \frac{-1}{2\sigma_k^2} \sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}^{(k)})^2 + \frac{-\lambda_k}{2\sigma_k^2} \sum_{j=1}^{p} |\beta_j^{(k)}| \right\} \\
\beta_1|\mathbf{Y}, \lambda_k, \sigma_k^2 &\sim N\left(\frac{1}{n_k} \sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}^{(k)}), \frac{\sigma_k^2}{n_k} \right) \\
\sigma_k^2|\mathbf{Y}, \lambda_k, \beta_0, \boldsymbol{\beta} &\sim IG\left(\frac{n_k}{2} + p - 1, \frac{\sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}^{(k)})^2 + \lambda_k \sum_{j=2}^{p} |\beta_j^{(k)}|}{2} \right) \\
\lambda_k|Y, \sigma_k^2, \boldsymbol{\beta} &\sim G\left(\alpha_0 + p - 1, \gamma_0 + \frac{\sum_{j=2}^{p} |\beta_j^{(k)}|}{2\sigma_k^2} \right),
\end{aligned}
$$

where $X_i^{(k)}$ is the i-th row of $X^{(k)}$.

Here, we are presenting a model that is decomposed in multiple regressions to handle the full matrix of ratings. The structure can be seen as a variation of [13] and [14], where linear regressions are used to model each column. Our model provides a full probabilistic construction that allows to do the same under a Bayesian approach. A natural limitation of this approach is the fact that we are modeling Normal variables when in the usual recommendation systems problems the ratings are discrete. However, [14] shows that linear models are well suitable even with few possible values for the ratings in the matrix. Also, many other authors have used the same structure to discrete ratings such as [16], [2] and [13]. Note as well that even though we are making an approximation, our approach still gives to the system a natural way of sorting the items after the prediction, which is our main goal.

The model is capturing the correlation between items and supposing that users are observed conditional independently. Note that this choice is not the most usual. In the recommendation system literature, usually the correlation between users is modeled, specially in the collaborative filtering approach, while the items are considered conditionally independent. This can be achieved by making the matrix of users correlation in (3) to be a full matrix of parameters and changing the matrix of correlations between the items to the identity.

As in [14], we are using the model to fill the missing data in the ratings matrix with estimates. Under our approach, one can access the predictive posterior distribution, and obtain point estimates from there. Also, each sample from the MCMC can be used as a recommendation, allowing the system to have varying recommendations even without new inputs from the users.

3 COMPUTATIONAL METHODS

Under the presented model construction, one can use MCMC methods to sample from the posterior distribution of the unknown model parameters. This approach brings all information needed to make inference about these parameters under the Bayesian paradigm. A natural disadvantage using this algorithm is the computational demand. In practice, some parameters such as the parametric vector $\boldsymbol{\beta}$ brings a big complexity to the algorithm when drawing from it, since it is requires inverting a large matrix and also drawing from a high dimensional normal distribution. Taking this into account, we present a semi-Bayesian approach for the estimation of the model parameters, which improves the computational efficiency of the method while achieving similar results when compared to the full Bayesian approach.

A Gibbs sampling algorithm (see [5]) is proposed where the elements in $\boldsymbol{\beta}$ are replaced by their maximum a posterior point estimation at every iteration of the algorithm. It is interesting to note, as pointed out by [12], that when we

specify a Laplace prior to the regression coefficients, the maximum a posterior is the same as under the LASSO, which allows us to keep the same interpretation of the model construction under this penalization.

The other model parameters are sampled from their full conditionals, and we also sample from the predictive posterior distribution. Algorithm 1 shows the full construction, given initial values to σ_k^2, λ_k and all missing values of $\mathbf{Y}$.

Algorithm 1 Maximum a Posterior Approach

(1) Set the $\beta_{1_{op}}^{(k)}$ and $\boldsymbol{\beta}_{op}^{(k)}$ from the optimization step where

$$\boldsymbol{\beta}_{op}^{(k)} = \underset{\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{p-1}}{\arg\min} \left\{ \sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}^{(k)})^2 + \lambda_k \sum_{j=2}^{p} |\beta_j^{(k)}| \right\} \tag{17}$$

$$\beta_{1_{op}}^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}_{op}^{(k)}) \tag{18}$$

(2) Draw $\sigma_k^2 | \boldsymbol{\beta}^{(k)}, \mathbf{Y} \sim IG\left(\frac{n_k}{2} + p, \frac{\sum_{i=1}^{n_k} (y_{ik} - X_i^{(k)} \boldsymbol{\beta}^{(k)})^2 + \lambda_k \sum_{j=2}^{p} |\beta_j^{(k)}|}{2} \right)$

(3) Draw $\lambda_k | \sigma_k^2, \boldsymbol{\beta}^{(k)} \sim G\left(p + \alpha_0, \frac{1}{2\sigma_k^2} \sum_{j=2}^{p} |\beta_j^{(k)}| + \gamma_0 \right)$

(4) Draw from the predictive posterior distribution.

This approach has the same advantage of sampling from the posterior distribution, as one can use each iteration as a full recommendation. Using the maximum a posterior estimation of $\boldsymbol{\beta}$ instead of sampling from their full conditional substantially improved the computational costs without causing deterioration of the results.

4 APPLICATION

In this section we present an application of the proposed methodology to a matrix of ratings obtained from the MovieLens dataset. It can be a challenge in the business to improve recommendation to a specific segment group. For our application we did two experimental applications. One segment considered movies that were rated by more than 25 users and a second segment considered movies that were rated by more than 100 users. The dataset was then split into train and test samples, with 75% of the data being used for training. The proposed methodology (MAP) was applied and 100 draws were obtained from the predictive posterior distribution, after a burn-in of 10 iterations and thinning of size 2. To evaluate the model, the mean of the predictive posterior distribution was used to calculate the RMSE and the Mean Absolute Error. The results are given in Table 1, where we compare MAP with ALS [16] and PMF [10].

Table 1. RMSE and Absolute Error of ALS, PMF and MAP algorthms under a dataset from MovieLens

	More than 25 users			More than 100 users		
	ALS	PMF	MAP	ALS	PMF	MAP
RMSE	0.862407	0.871272	1.044578	0.836530	0.849594	0.921344
Mean Absolute Error	0.668212	0.676056	0.771120	0.645152	0.651407	0.701926

It can be seen that even with worst results, MAP has competitive metrics values, when comparing with both ALS and PMF. Besides that, MAP showed a significant improvement when used in segments that have less sparsity levels.

These segments, such as heavy users or items that are usually consumed by many users, reduce the sparsity of the ratings matrix and are a particular focus of companies. One natural advantage of our approach is drawing from the posterior distribution the parameters. With this, other metrics can be used to sort the items to be recommended, taking into account not only a mean or median estimation, but also the variation estimated for each item. Another suggestion is that each draw from the predictive posterior distribution can be used as a recommendation. Therefore, we took 5 random recommendations structures and evaluate the RMSE in each one of them. Table 2 shows the error achieved in each structure.

Table 2. RMSE for 5 random samples from the predictive posterior distribution for items rated by more than 25 users

Sample	RMSE
1	1.079552
2	1.044665
3	1.046287
4	1.066011
5	1.039273

The obtained results show that our approach achieved overall a good performance when considering point estimation. It is interesting to note, however, that a recommendation system based on point estimates alone would clearly suggest a fixed ranking of movies. We propose to make recommendations at every iteration of the algorithm, instead. Since we draw several samples from the predictive posterior distribution, the predicted ratings vary for a given user i, allowing new movies to be recommended at different steps of the algorithm. We believe this diversification is a desirable property of our methodology as users may get unusual recommendations from time to time, which may not only be welcome by the users themselves but also help the learning mechanism of the algorithm.

5 CONCLUSION

Since the main interest of the proposed model is to sample from the predictive posterior distribution, a semi-Bayesian approach was presented using Bayesian optimization steps in the algorithm. Another important aspect that we must take into account in this field of application is the computational time that is required for the estimation of the model and obtaining of the final product of interest, which is the recommendation. Instead of sampling from the regression parameters $\boldsymbol{\beta}$ at every iteration of the MCMC algorithm, we use their maximum a posterior estimates at each step. This substantially improved the computational costs without causing deterioration of the results.

A next step in our research is to consider cases where new users and new movies are introduced into the system. It is a challenge to recommend movies to new users, when we have no information about their preferences. Also, when new movies are introduced, to whom must they be recommended? With our approach, the vectorization of the ratings matrix allows one to see each row as a rating to be estimated. In particular, a new latent vector representing a new user or a new item can be estimated and the system will be ready for making recommendations for this users.

When working under a Bayesian approach, one good advantage is the possibility of assigning prior knowledge to the parameters. If there are metadata available about the items, one possible future work would be to try to find correlation or similarity between the items and input it in the prior of the regressors parameters. With this approach, the system will use both information: the users preferences and the available information about the items.

REFERENCES

[1] Charu C. Aggarwal and Srinivasan Parthasarathy. 2001. Mining Massively Incomplete Data Sets by Conceptual Reconstruction. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/502512.502543

[2] Sungjin Ahn, Anoop Korattikara, Nathan Liu, Suju Rajan, and Max Welling. 2015. Large-scale Distributed Bayesian Matrix Factorization using Stochastic Gradient MCMC. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 9–18.

[3] Barry C Arnold and S James Press. 1989. Compatible Conditional distributions. *J. Amer. Statist. Assoc.* 84, 405 (1989), 152–156.

[4] Dani Gamerman and Hedibert F Lopes. 2006. *Markov chain Monte Carlo: stochastic simulation for Bayesian Inference*. Chapman and Hall/CRC.

[5] Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), 721–741.

[6] Arjun K Gupta and Daya K Nagar. 2018. *Matrix Variate Distributions*. Chapman and Hall/CRC.

[7] F Maxwell Harper and Joseph A Konstan. 2016. The Movielens Datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2016), 19.

[8] Richard Arnold Johnson, Dean W Wichern, et al. 2002. *Applied Multivariate Statistical Analysis*. Vol. 5. Prentice hall Upper Saddle River, NJ.

[9] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 1297–1305. http://papers.nips.cc/paper/5985-efficient-thompson-sampling-for-online-matrix-factorization-recommendation.pdf

[10] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[11] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of dimensionality reduction in recommender System-a case study*. Technical Report. Minnesota Univ Minneapolis Dept of Computer Science.

[12] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.

[13] Zhonghang Xia, Yulin Dong, and Guangming Xing. 2006. Support Vector Machines for Collaborative Filtering. In *Proceedings of the 44th annual Southeast regional conference*. 169–174.

[14] Tong Zhang and Vijay S Iyengar. 2002. Recommender Systems using Linear classifiers. *Journal of machine learning research* 2, Feb (2002), 313–334.

[15] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive Collaborative Filtering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management* (San Francisco, California, USA) *(CIKM '13)*. ACM, New York, NY, USA, 1411–1420. https://doi.org/10.1145/2505515.2505690

[16] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel Collaborative Filtering for the netflix prize. In *International conference on algorithmic applications in management*. Springer, 337–348.