Movie Recommendation System via Markovian Factorization of Matrix Process

A MINI PROJECT REPORT

submitted by

DINTO DAVI T

(TCR19CSCE08)

to

the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering

Government Engineering College

Thrissur

APRIL 2020

DECLARATION

I undersigned hereby declare that the project report ”Collaborative Filtering-Based Movie Recommendation System”, submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Assist.Prof. RAHAMATHULLA K**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place Signature

Date DINTO DAVI T.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT ENGINEERING COLLEGE,THRISSUR

CERTIFICATE

This is to certify that the report entitled **"Collaborative Filtering-Based Movie Recommendation System",** submitted by **DINTO DAVI T** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering is a bonafide record of the project carried out by his under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor(s) **External Supervisor(s)**

PG Coordinator **Head of the Department**

ACKNOWLEDGEMENT

I give honour and praise to the lord who gave me wisdom and enabled me to complete this project successfully.

I would like to thank the Department of Computer Science and Engineering, Government Engineering College, Thrissur, for giving me the opportunity to present this project.

I would also like to thank **Prof. VIPIN KUMAR K S**, HOD, Deparment of Computer Science and Engineering for rendering his help and guidance during the completion of the project.

I would like to express my deep gratitude to our PG Co-ordinator **Dr. Swaraj K P** for his timely suggestions and encouragement and guiding through the processes involved in the presentation of the project and the preparation of the report.

I would like to thank our project Co-ordinator **Dr.Shreelekshmi.R and assist.prof.Mumthas** for their sincere support throughout the project presentation.

I would also like to thank my guide **Assist.Prof. RAHAMATHULLA K** for rendering his help and guidance throughout the project presentation.

Last, but not the least, I express my heartfelt thanks to my friends and parents for their support and encouragement during this endeavor.

ABSTRACT

Collaborative filtering method was widely used in the recommendation system. This method was able to provide recommendations to the user through the similarity values between users.Recommendation systems which are becoming a norm for consumer industries such as books, music, clothing, movies, news articles, places, utilities, etc.These systems collect information from the users to improve the future suggestions.During the last few decades, with the rise of Youtube, Amazon, Net ix and many other such web services, recommender systems have taken more and more place in our lives.From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called user-item interactions matrix".The main advantage of collaborative approaches is that they require no information about users or items and, so, they can be used in many situations. This project created using Markovian factorization of matrix process (MFMP) model.On one hand, MFMP models, such as timeSVD++, are capable of capturing the temporal dynamics in the dataset, and on the other hand, they also have clean probabilistic formulations, allowing them to adapt to a wide spectrum of collaborative filtering problems. The experimental study using MovieLens dataset demonstrates that the two models, although simple and primitive, already have comparable or even better performance than timeSVD++ and a standard tensor factorization model.

Keywords: *Recommender system, collaborative filtering, matrix factorization.*

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

A typical collaborative filtering problem is the rating prediction problem prescribed by a dataset consisting of N users, M movies, and a collection of user ratings R_{ij}'s each given by some user i on some item j.In reality, most item have only been rated by a fraction of the users and most users have only rated a fraction of the movies.As a consequence, when organizing the observed rating R_{ij}'s in an M×N matrix R, a large fraction of the matrix entries are missing.The objective of collaborative filtering in this setting is to predict the missing entries of R based on the observed ratings.

Probabilistic Matrix Factorization(PMF) assumes that there is a latent space of dimension D,a number much smaller than M and N, from which each user i is associated with a user feature vector U_i and each movie is associated with a movie feature vector V_j. The rating of movie j given by user i is then modeled as the inner product of the two vectors, $U_i^T V_j$, subject to additive Gaussian noise.

Simple and elegant as it is, PMF appears surprisingly effective in practice and has since been applied to many application domains, which include to generate annotations for images, to recommend products for social network users, and to recommend programs for IPTV users.Inspired by its great success, various extensions have been made to PMF models to date,among which two directions are particularly noteworthy.One direction applies a probabilistic framework similar to PMF for completing higher-dimensional data arrays, or tensors; this has led to various "tensor factorization" models. The other direction discards the assumption that the PMF model is static across time and allows the model to evolve over time in order to capture the dynamics of ”concept drift” in collaborative filtering; this has led to the timeSVD++ algorithm for collaborative filtering.

In the context of predicting movie ratings, when the timestamps of the collected ratings are available, the dataset presents itself as a three-dimensional array, or a three-mode tensor. In this case, both tensor factorization models and timeSVD++ can be used as solvers for the rating prediction (or ”tensor completion”) problem. In fact, depending on the dataset and its intrinsic temporal dynamics, the two classes of solvers may indeed be close competitors against each other. One advantage of tensor factorization models is their clean and well-principled probabilistic formulations, along a line very similar to that of PMF. These models are however too general: they are not specifically designed to capture concept drift over time and they assume no correlation or dependency structure specific for temporal dynamics. On the other hand, the timeSVD++ algorithm is specifically crafted for capturing concept drift in rating prediction problems.

we would like to develop a family of models or a modeling methodology which captures concept drift in collaborative filtering and which also has a probabilistic formulation. The model family we present in this paper is termed Markovian Factorization of Matrix Process (MFMP). Using the movie rating prediction problem as a running example throughout the paper, the methodology of this work models the rating matrix R as a matrix- valued random process $\{$R(t)$\}$. We associate with each user i a user feature process $\{U_i(t)\}$ and with each movie j a movie feature process $\{V_j(t)\}$. The (i,j) entry of matrix process $\{$R(t)$\}$, namely random process $\{R_{ij}(t)\}$, is then modeled as the “inner-product process” $\{U_i(t)^T V_j(t)\}$ subject to probabilistic perturbation. Markovian structures are imposed on the latent processes (namely, all $\{U_i(t)\}$’s and all $\{V_j(t)\}$’s) in order to capture the temporal dynamics of $\{$R(t)$\}$. These assumptions define the family of MFMP models.
To be more concrete, we present two simple examples in this model family, which we call the first-order MFMP and the second-order MFMP. The two models as-

sume the latent processes to be respectively first-order and second order Gaussian Markov processes and probabilistic perturbation on the inner-product processes is also assumed to be Gaussian, taking effect additively.Using time-stamped movie rating dataset from MovieLens, we show that the algorithms developed from both models already have comparable or even better performance than timeSVD++ and a standard tensor factorization algorithm.Using Markov processes or Hidden Markov Models to model time series is not at all a new idea. In fact, such an approach has prevailed in the area of statistical signal processing for the past decades. In the area of collaborative filtering, a Hidden Markov Model was also developed for interpreting users' blog-reading behavior and making article recommendations .

CHAPTER 2

LITERATURE SURVEY

2.1 RECOMMENDER SYSTEM

The general goal of recommender systems is to assist potential buyers in discovering items, such as products or information.Collaborative filtering has been successfully exploited by many systems to predict the ratings by aggregating the similar experiences. Traditional recommendation models,such as item-based [2], user-based [3] and hybrid algorithms [4] all have been shown the capability of providing higher quality recommendations in various domains [5], [6]. With the increasing number of users and items emerged, the scalability [7], [8], the efficiency [9], [10] and the stability [11] have been studied to extend the traditional recommendation algorithms to meet the huge computation requirement.
Matrix factorization or SVD is one of such model-base techniques and it has been successfully adopted by the application domains such as document clustering, facial recognition and collaborative filtering. For the collaborative filtering, it becomes another popular modeling approach since their success in Netflix challenge. Moreover, traditional neighbor based collaborative filtering is combined with MF model [12] for improving the recommendation accuracy. These existing models are however not specifically designed to capture concept drift over time and they assume no correlation or dependency structure specific for temporal dynamics. In this article, we propose a generic model by incorporating hidden Markov models into matrix factorization and provide algorithms for solving the concept drifting prob-

lem in recommender system. Context-aware recommendation algorithms [13] are such technique which used to characterize the correlation between users' dynamic preferences and their contexts.

2.2 USER PREFERENCE MODELS

User preference modeling and discovering is one of the important problem in recommender system. Existing studies have presented many potential ways to increase the recommendation performance by incorporating the user preferences explicitly or implicitly. For example, Wu et al. [14] and Michelson and Macskassy [15] aim at extracting keywords from user generated contents to represent their explicit user interests.
The matrix factorization [16], [17] and hierarchical matrix factorization [18], [19], [20] can also be seen as a model for discovering and making use of the user implicit preferences. The decomposed user low-rank matrix can be seen as a representation of user preferences.The matrix factorization model can discovering the implicit preferences and overcoming the sparsity problem at the same time. Also, the approaches based on this model promote the development of the recommender system. These existing studies merely focus on the static modeling of user preferences. However, in reality, user behavior or requirements might be changing over time. This fact makes it possible to model the user preference dynamics together with the traditional collaborative filtering approach

2.3 MARKOV MODEL

As the objective of this paper is to promote a general statespace model to deal with temporal dynamics in collaborative filtering, Markov processes naturally arise as they are commonly used for constructing the dependencies between adjacent temporal slots. There have been existing previous works that take both hidden Markov models and collaborative filtering into consideration when building models. However, some of these models, such as [21], study the sequences between the items purchased by users. Others, such as [22], model the latent processes as first-order Markov processes and which give rise to Kalman filters (KF). The MFMP model of this paper goes beyond first-order Markov process and in fact the latent processes in MFMP model family can be a Markov process of arbitrary order and non-Gaussian. In addition, the main contribution of this work is not advocating one or two specific models and testing them for some specific dataset.

The previous works combining KF with MF may be viewed as specific examples of the MFMP family and may be seen as a testimony of the usefulness of the MFMP modeling framework. Noting that none of the previous works considers higher-order or more general latent Markov processes and our MFMP models are essential classical state-space models (hidden Markov models) adapted to matrix factorization settings.

CHAPTER 3

MARKOVIAN FACTORIZATION OF MATRIX PROCESS MODELS

3.1 GENERAL FORMULATION

Let {1,2,..,M} index the set of items (e.g. movies) and {1,2,..,N} index the set of users. Let {0,1,2..,T} index the set of discrete time points. There is a latent space $R^D$of dimension D with D much smaller than M and N.At every time point t ϵ {0,1,2..T }, associate with each user i a vector U_i(t)ϵ R^D and associate each item j a vector V_j(t) ϵ R^D.Intuitively V_j(t) may be regarded as the latent feature of item j at time t and U_i(t) may be regarded as the "weighting" vector of user i at time t.We assume that all user feature processes {U_i(t)}'s and all item feature processes {V_j(t)}'s are mutually independent.Collectively, for any fixed t, we denote {$U_i(t)$: i=1,2,..,N} and {V_j(t): i=1,2,...,M} by D $\times$ N matrix U(t) and D $\times$ M matrix V(t) respectively.

We model every {U_i(t)} process and every {V_j(t)} process both as Markov processes, and model, for each (i,j) pair, R_{ij}(t) to only depend on the inner product of $U_i(t)^T V_j(t)$.Depending on further specification of the processes {U_i(t)}'s and {V_j(t)}'s and the dependency of R_{ij}(t) on ($U_i(t)^T V_j(t)$), various models may be constructed. such a model called as Markovian Fractorization of Matrix Process (MFMP) model.

3.2 FIRST-ORDER MFMP

Given U_i(t) and V_j(t), the (i,j) entry of the matrix R(t) is modeled as

$$R_{i,j}(t) = U_i(t)^T V_j(t) + Z_{ij}(t), \tag{1}$$

where $Z_{ij}(0)$,$Z_{ij}(1)$,... $Z_{ij}(T)$ are i.i.d. Gaussian random variables with zero mean and variance $\sigma^{\mathbf{2}}$. Then it follows that the distribution of R_{ij}(t) conditioned on U_i(t) and V_j(t) is

$$p(R_{ij}(t)|U_i(t), V_j(t)) = \mathcal{N}(R_{ij}(t)|U_i(t)^T V_j(t), \sigma^2) \tag{2}$$

where we have used the notation $\mathcal{N}(x|\mu, \sum)$ to denote the Gaussian probability density function with variable x, mean vector μ and covariance matrix $\sum$. Note that in (2), the Gaussian density is univariate, in which the covariance matrix reduces to a scalar value.For simplicity, we denote matrix processes {U(t)}, {V(t)} and {R(t)} over time t=0,1,2,..,T by U,V and R respectively. At the time origin t=0 , we model U_i(0) to follow a zero-mean spherical Gaussian distribution for each user i , namely,

$$p(U_i(0)) = \mathcal{N}(U_i(0)|0, \Sigma_U^2 \mathbf{I}) \tag{3}$$

Likewise, at t=0 , we model V_j(0) to follow another zero-mean spherical Gaussian distribution for each item j , namely,

$$p(V_j(0)) = \mathcal{N}(V_j(0)|0, \Sigma_V^2 \mathbf{I}) \tag{4}$$

The evolution of U_i across time for each user i is modeled as

$$U_i(t+1) = U_i(t) + X_i(t), \tag{5}$$

where $\{X_i(t)\}$ is an i.i.d. Gaussian process with each X_i(t) drawn from distribution $N(X_i(t)|0, \sigma_U^2 I)$. Essentially X_i(t) models the drift of feature vector of user i at time t.

Similarly, the evolution of V_j across time for every item is modeled as

$$V_j(t+1) = V_j(t) + Y_j(t), \tag{6}$$

where $\{Y_j(t)\}$ is an i.i.d. Gaussian process with each $Y_j(t)$ is drawn from distribution $N(Y_j(t)|0, \sigma_V^2 I)$. Apparently $Y_j(t)$ models the drift of feature vector of item j at time t. Under these assumptions, it is evident that both U_i process and V_j process are Gaussian Markov processes. Moreover processes $U_i, V_j and R_{ij}$ form a Hidden Markov Model with $(U_i(t), V_i(t))$ being the latent state of the underlying Markov chain at time t and $R_{ij}(t)$ being the observed variable at time t. This is best explained using the Bayesian network in Figure 3.1.

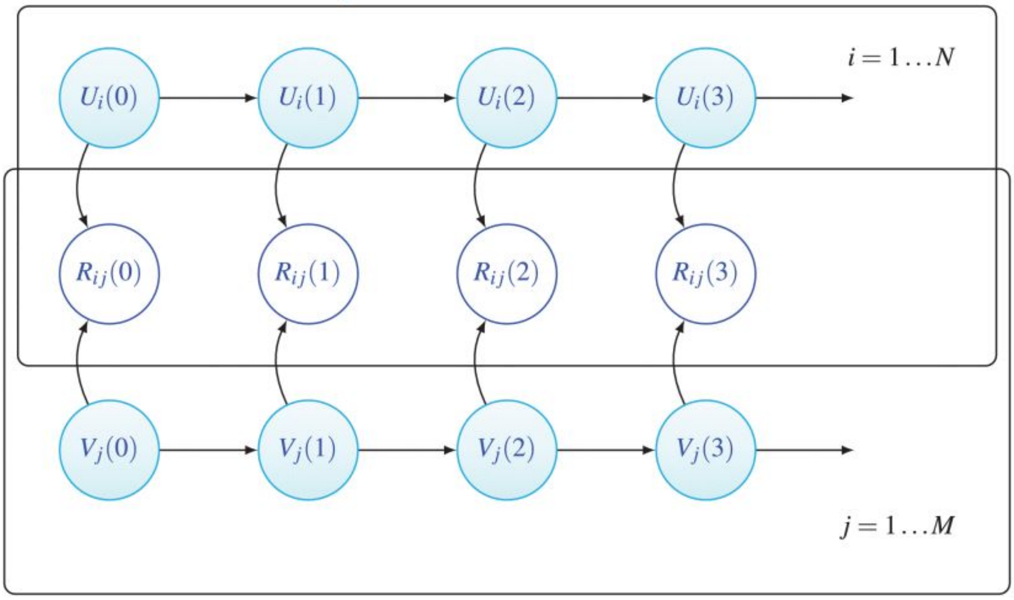


Figure 3.1: First-order MFMP model.

3.3 SECOND-ORDER MFMP

In this model, instead of modeling the drifting terms $\{X_i(t)\}$ and $\{Y_j(t)\}$ being independent across time, we model them as (first order) Gaussian Markov processes.

$$p(X_i(0)) = \mathcal{N}(X_i(0)|0, \sigma_U^2 \mathbf{I}) \tag{7}$$

$$X_i(t+1) = a_U X_i(t) + \Gamma_i(t), \tag{8}$$

where for each i , $\{\Gamma_i(t)\}$ is an i.i.d. Gaussian process with zero mean and variance $b_U\sigma_U^2$, and all such processes are independent across all i 's; similarly,

$$p(Y_j(0)) = \mathcal{N}(Y_j(0)|0, \sigma_V^2\mathbf{I}) \tag{9}$$

$$Y_j(t+1) = a_V Y_j(t) + \Delta_j(t), \tag{10}$$

where for each j , $\{\Delta_j(t)\}$ is an i.i.d. Gaussian process with zero mean and variance $b_V\sigma_V^2$, and all such processes are independent across all j 's. Modifying the first-order MFMP model according to Equations (7) to (10) specifies the second-order MFMP model.

This gives rise to the graphical model in Figure 3.2.

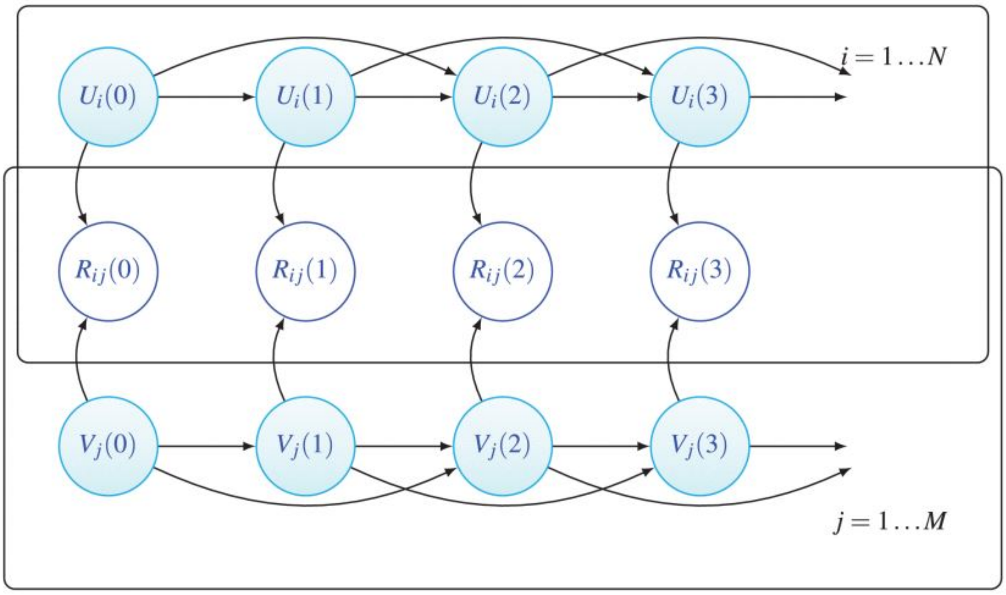


Figure 3.2: Second-order MFMP model with drifting processes marginalized out.

It is easy to see from the figure that the latent processes U and V in this model both are second-order Markov processes.

The first-order and second-order MFMP models presented above are merely the simplest examples in the MFMP model family. Depending on the temporal dynamic nature of the dataset, one may impose higher-order Markovian structure on the latent processes. For data of other types, one may also consider distributions beyond the Gaussian family. The probabilistic dependency of $R_{ij}(t)$ on the inner product

$U_i(t)^T V_j(t)$ may also be made arbitrary. In fact, a Hidden Markov Model has been recently presented for interpreting users' blog reading behavior and making article recommendations. The model thereof consists of only one latent non-Gaussian Markov Matrix process, which serves as the user feature process. One may regard this model as a special case of the MFMP model family, in which the process V(t) is trivialized.

CHAPTER 4

EVALUATION METRICS

Experimental study is performed to evaluate the effectiveness of the proposed first-order and second-order MFMP models in the prediction of movie ratings.

4.1 DATASETS AND EVALUATION METRIC

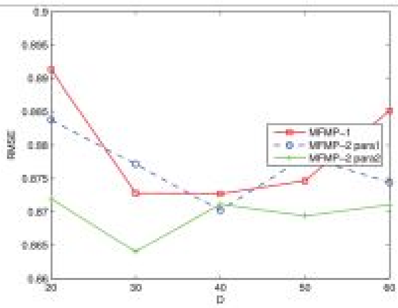
Two datasets we use in the experiments are from MovieLens- 1M and MovieLens-20M.MovieLens-1M data set contains 1 million ratings from 6,000 users on 4,000 movies. MovieLens-20M data set contains and 20 million ratings from 138,000 users on over 27,000 movies respectively. These two dataset also contain the time-stamp of when each user rated the movies. Movies in MovieLens-1M data set were rated in a 1040 days period and in MovieLens-20M data set were rated in a 20 years period. The ratings are on the scale of 1 to 5. As is standard, root mean squared error (RMSE) and Precision@10 are used as the evaluation metric.

4.2 PARAMETER SELECTION AND OVERALL PERFORMANCES OF MFMP MODELS

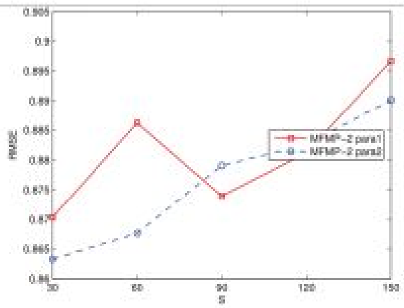
Existing matrix factorization methods merely provide the functionality of predicting missing values of a matrix. However, future value prediction is also impor-

tant for many cases. As the proposed model characterizes the evolution of users and items, it naturally support the future prediction. Such, in this empirical study, the performances of future rating prediction is our focus to be examined.
In the experiments, we randomly select 80% of time-stamped ratings as the training set and take the remaining set of ratings as the testing set. Time is divided into slots of duration S days, which we let vary in our experiments.For MovieLens-1M dataset, the parameters for the first-order MFMP are chosen as ρu=ρv=0.05,λu=λv=100 . For MovieLens-20M dataset, the parameters for the first-order MFMP are chosen as ρu=ρv=0.01,λu=λv=200 . Noting the probabilistic meanings of the parameters, these parameter settings allow relatively large variation across the elements of any given latent (user or movie) feature vector but only allows small drifts across time.
For the second-order MFMP on MovieLens-1M dataset, two sets of parameters are chosen: Para1=$\{a_U = 0.3, b_U = 0.5, a_V = 0.1, b_V = 0.08\}$ and Para2=$\{a_U = 0.35, b_U = 0.1, a_V = 0.8, b_V = 0.5\}$. For the second-order MFMP on MovieLens-20M dataset, two sets of parameters are chosen: Para1=$\{a_U = 0.5, b_U = 0.8, a_V = 0.05, b_V = 0.1\}$ and Para2=$\{a_U = 0.7, b_U = 0.15, a_V = 1.1, b_V = 0.65\}$. With Para1 setting, the choice aV as a small value eliminates the correlation between consecutive drifts on movie feature vectors, which reduces the $\{$V(t)$\}$ process to a first-order Markov process, and the choice bV as a small value allows only small drifts in movie feature vectors across time. With Para2, the model allows modest correlation in both user feature drifts and in movie feature drifts.

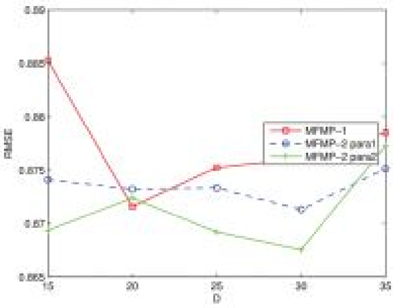
Figure 4.1 demonstrate performances of the first-order MFMP model (MFMP-1) and second-order MFMP model (MFMP-2) with varying settings of latent space dimension D on MovieLens-1M and MovieLens-20M respectively. In this figure 4.1, it appears that the second-order MFMP models overall perform better than the first-order model. Comparing the two second-order MFMP models, setting Para1 appears to perform better than setting Para2 when the choice of D is relatively small and the two settings perform similarly for larger values of D. According to the evaluation results, we choose D D 30 and S D 30 for MovieLens-1M dataset and D D 30 and S D 90 for MovieLens-20M dataset for further experiments. The two parameter settings of the second-order MFMP model are further investigated for various choices of time slot size S and the results are plotted in Figure 4.1. The two settings result in similar performances. It appears that with small values of S, Para1 performs better, and with larger values of S, Para2 performs better. This makes it difficult to conclude whether in this dataset correlation indeed exists across consecutive drifts of movie features.



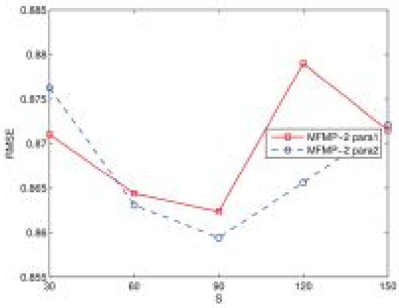
(1) Performance of MFMP models on MovieLens-1M dataset for varying values of latent space dimension D. $S = 30$ days.



(2) Performance of the second-order MFMP models on MovieLens-1M dataset for various choices of time slot size S. $D = 30$.



(3) Performance of MFMP models on MovieLens-20M dataset for varying values of latent space dimension D. $S = 90$ days.



(4) Performance of the second-order MFMP models on MovieLens-20M dataset for various choices of time slot size S. $D = 30$.

Figure 4.1: The choices of latent space dimension D and time slot size S . (1) Performance of MFMP models on MovieLens-1M dataset for varying values of latent space dimension D . S=30 days. (2) Performance of the second-order MFMP models on MovieLens-1M dataset for various choices of time slot size S . D=30 . (3) Performance of MFMP models on MovieLens-20M dataset for varying values of latent space dimension D . S=90 days. (4) Performance of the second-order MFMP models on MovieLens-20M dataset for various choices of time slot size S . D=30.

CHAPTER 5

CONCLUSION

More general models beyond those relying on Gaussian distributions and additive drifts should be considered when there is significant presence of such phenomenon in the datasets.MFMP models are a rich family of probabilistic models that marry the framework of PMF with the framework of Hidden Markov Models. By varying the order of the latent Markov processes, the involved distributions and the dependency of observation on the latent process, a large variety of temporal dynamical models can be constructed for collaborative filtering problems.One possible extension of this study is to include the textual information associated with the rating to handle the “cold-start” or “early-voter” problems.PMF may reduce to PCA when trivializing one of the latent matrix factors to a vector. This chain of hierarchy motivates an interest in mapping out the two MFMP models of this project in the big picture of these factorized models.

.

REFERENCES

[1] R Zhang, Y Mao. **”Movie Recommendation via Markovian Factorization of Matrix Processes”**- IEEE Access, 2019.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, **”Item-based collaborative filtering recommendation algorithms,”** in Proc. ACM 10th Int. Conf. World Wide Web (WWW), New York, NY, USA, 2001, pp. 285-295, doi: 10.1145/371920.372071.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, **”GroupLens: An open architecture for collaborative filtering of netnews,”** in Proc. ACM Conf. Comput. Supported Cooperat.Work (CSCW), New York, NY, USA, 1994, pp. 175-186, doi: 10.1145/192844.192905.

[4] R. Burke, **”Hybrid recommender systems: Survey and experiments,”** User Model. User-Adapted Interact., vol. 12, no. 4, pp. 331-370, Nov. 2002, doi: 10.1023/A:1021240730564.

[5] G. Adomavicius and A. Tuzhilin, **”Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,”** IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734-749, Jun. 2005.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez, **”Recommender systems survey,”** Know.-Based Syst., vol. 46, pp. 109-132, Jul. 2013, doi: 10.1016/j.knosys.2013.03.012.

[7] G. Takács, I. Pilászy, B. Németh, and D. Tikk, **”Scalable collaborative filtering approaches for large recommender systems,”** J. Mach. Learn. Res., vol. 10, pp. 623-656, Mar. 2009, doi: 10.1145/1577069.1577091.

[8] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q.-S. Zhu, **”A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method,”** IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 3, pp. 579-592, Mar. 2016.

[9] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, **”An efcient non-negative matrixfactorization- based approach to collaborative filtering for recommender systems,”** IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1273-1284, May 2014.

[10] X. Luo, M. C. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, and H. Leung, **”An efficient second-order approach to factorize sparse matrices in recommender systems,”** IEEE Trans. Ind. Informat., vol. 11, no. 4, pp. 946-956, Aug. 2015.

[11] G. Adomavicius and J. Zhang, **“Stability of recommendation algorithms,”** ACM Trans. Inf. Syst., vol. 30, no. 4, pp. 23-1-23-31, Nov. 2012, doi: 10.1145/2382438.2382442.

[12] Y. Koren, **“Factorization meets the neighborhood: A multifaceted collaborative filtering model,”** in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), New York, NY, USA, 2008, pp. 426-434.

[13] G. Adomavicius and A. Tuzhilin, **“Context-aware recommender systems,”** in Recommender Systems Handbook. Boston, MA, USA: Springer, 2011, pp. 217-253.

[14] M. Michelson and S. A. Macskassy, **“Discovering users’ topics of interest on Twitter: A first look,”** in Proc. ACM4thWorkshop Anal. Noisy Unstructured Text Data, 2010, pp. 73-80.

[15] J. Zhang, Y. Lin, M. Lin, and J. Liu, **“An effective collaborative filtering algorithm based on user preference clustering,”** Appl. Intell., vol. 45, no. 2, pp. 230-240, Sep. 2016.

[16] R. R. Salakhutdinov and A. Mnih, **"Probabilistic matrix factorization,"** in Advances in Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 2008, pp. 1257-1264.

[17] Y. Koren, R. Bell, and C. Volinsky, **"Matrix factorization techniques for recommender systems,"** IEEE Comput., vol. 42, no. 8, pp. 30-37, Aug. 2009.

[18] H. Shan, J. Kattge, P. Reich, A. Banerjee, F. Schrodt, and M. Reichstein, **"Gap filling in the plant kingdom-Trait prediction using hierarchical probabilistic matrix factorization,"** in Proc. 29th Int. Conf. Mach. Learn. (ICML), Scotland, U.K., Jun./Jul. 2012, pp. 1303-1310.

[19] X. Wang, W. Pan, and C. Xu, **"HGMF: Hierarchical group matrix factorization for collaborative recommendation,"** in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM), Shanghai, China, Nov. 2014, pp. 769-778.

[20] E. Zhong, W. Fan, and Q. Yang,**"Contextual collaborative filtering via hierarchical matrix factorization,"** in Proc. 12th SIAM Int. Conf. Data Mining, Anaheim, CA, USA, Apr. 2012, pp. 744-755.

[21] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, **"Factorizing personalized Markov chains for next-basket recommendation,"** in Proc. ACM 19th Int. Conf. World Wide Web (WWW), New York, NY, USA, 2010, pp. 811-820.

[22] N. Sahoo, P. V. Singh, and T. Mukhopadhyay, **"A hidden Markov model for collaborative filtering,"** MIS Quart., vol. 36, no. 4, pp. 1329-1356, Dec. 2012.