

Matrix Factorization with Time Dynamics

Umang Patel (ujp2001) and Ilambharathi Kanniah (ik2342)

Columbia University

25th April 2014

Introduction

- ▶ Movie Recommendation is a very well looked into problem. (e.g. Netflix Prize Challenge)
- ▶ In this project, we are exploring the applications of time dynamics of user in movie recommendation problem.
- ▶ Existing solutions include timeSVD, target tracking in recommender space (no collaborative filtering) [NBB10]

timeSVD

timeSVD addresses temporal dynamics with factors drifting from a central time unlike a general formulation and hence can only handle limited temporal structure.

Idea

- ▶ Collaborative Filtering
- ▶ Singular Value Decomposition (SVD) - No time factor
- ▶ Probabilistic Matrix Factorization (PMF) [SM08b] - MAP estimate, but again no time factor.
- ▶ Time Dynamics of Users
- ▶ Hidden Markov Model [SSM12] - To include user dynamics
- ▶ Kalman Filter (KF) [Fle10] - Includes process and measurement noise
- ▶ Kalman Filter (KF) in Expectation-Maximization (EM) [TR06] [SVS11]
- ▶ Application of Movie Recommendation using KF in EM

Problem Formulation

N - Number of Users, M - Number of Movies

$O \in \mathbb{R}^{N \times M}$ (Sparse Matrix of Entries)

Matrix Factorization (MF) - $U \in \mathbb{R}^{N \times K}$ and $V \in \mathbb{R}^{M \times K}$ where there are K latent factors.

Then, the preference (ratings) matrix,

$$O = UV^T$$

Given K,

$$\min_{U, V} \sum_{(i, j) \in O} (o_{ij} - u_i v_j^T)^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_F^2$$

for $i=1$ to N and $j=1$ to M

Bayesian Probabilistic Matrix Factorization

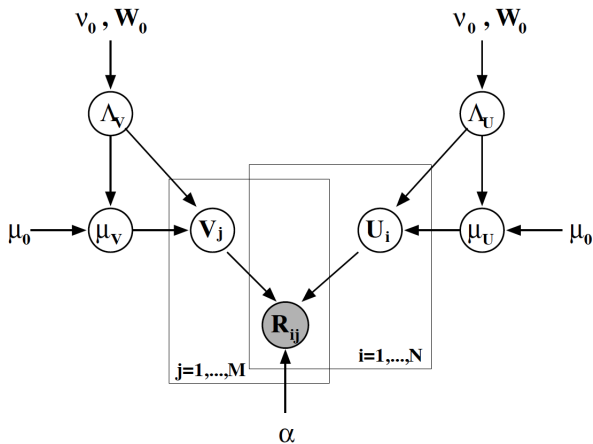


Figure 1 : Plate Diagram [SM08a]

Bayesian Probabilistic Matrix Factorization

$$p(U|\mu_U, \Lambda_U) = \prod_{i=1}^N N(u_i|\mu_U, \lambda_U^{-1})$$

$$p(V|\mu_V, \Lambda_V) = \prod_{i=1}^M N(v_i|\mu_V, \lambda_V^{-1})$$

$$\theta_U = \{\mu_U, \Lambda_U\}, \theta_V = \{\mu_V, \Lambda_V\} \text{ (has Gaussian-Wishart prior)} \quad (1)$$

$$\begin{aligned} p(\theta_U|\theta_0) &= p(\mu_U|\Lambda_U)p(\Lambda_U) \\ &= N(\mu_U|\mu_0, (\beta_0\Lambda_U)^{-1})W(\Lambda_U|W_0, \nu_0) \end{aligned}$$

$$\begin{aligned} p(\theta_V|\theta_0) &= p(\mu_V|\Lambda_V)p(\Lambda_V) \\ &= N(\mu_V|\mu_0, (\beta_0\Lambda_V)^{-1})W(\Lambda_V|W_0, \nu_0) \end{aligned}$$

Solution

Extension to KF in EM - Factorization via Learning

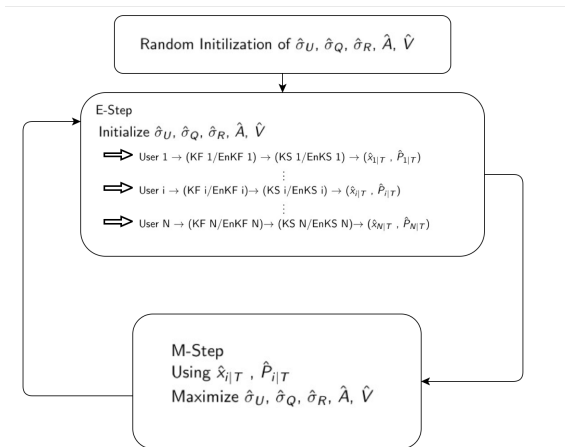


Figure 2 : Process Diagram

Solution

Extension to KF in EM - Factorization via Learning

Expectation - Maximization Algorithm

Input : Sparse Observation Matrix Y E-Step

- Initialize User States from $\mathcal{N}(0, \hat{\sigma}_U^2)$
- Process Noise from $\mathcal{N}(0, \hat{\sigma}_Q^2)$ and Measurement Noise from $\mathcal{N}(0, \hat{\sigma}_R^2)$
- Forward Pass - Kalman Filter, Backward Pass - RTS (Rauch-Tung-Striebel) Smoother (since all observations w.r.t users and time are available) using the sparse Observation Matrix Y .

M-Step

- Maximize Log Likelihood to get maximized estimates for $\hat{\sigma}_U^2$ (Variance of Initial States)
- $\hat{\sigma}_Q^2$ (Variance for Process Noise Model), $\hat{\sigma}_R^2$ (Variance for Measurement Noise Model)
- \hat{A} - Process Transition Matrix
- \hat{V} - Observation Matrix (Movie Factor Matrix).

Output : The estimated User-States \hat{U} and estimated complete Observation Matrix \hat{Y}

Algorithm: Expectation Maximization Algorithm with E-Step having Kalman Filter and Smoother

Initial Clustering of User Matrix and then running KF

Matrix Factorization with Initial User Clustering

- Input : Sparse Observation matrix Y
- Form initial clusters by decomposing initial observation Y_{true} with SVD. Set of Clusters, SC having NC clusters.
- For each cluster in SC set of clusters.
 - E-Step
 - * Initialize User States from $\mathcal{N}(0, \hat{\sigma}_U^2)$
 - * Process Noise from $\mathcal{N}(0, \hat{\sigma}_Q^2)$ and Measurement Noise from $\mathcal{N}(0, \hat{\sigma}_R^2)$
 - * Forward Pass - Ensemble Kalman Filter, Backward Pass - Ensemble Kalman Smoother (since all observations w.r.t users and time are available) using the sparse observation matrix Y
 - M-Step
 - * Maximize Log Likelihood to get maximized estimates for $\hat{\sigma}_U^2$ (Variance of Initial States)
 - * $\hat{\sigma}_Q^2$ (Variance for Process Noise Model), $\hat{\sigma}_R^2$ (Variance for Measurement Noise Model)
 - * \hat{A} - Process Transition Matrix
 - * \hat{V} - Observation Matrix (Movie Factor Matrix).
- Combine the estimated observations for all users in the clusters and movies(items) into estimated complete \hat{Y} .
- Output the estimated complete \hat{Y} .

Algorithm: Initial User Clustering with Expectation Maximization

Notation

- ▶ $\hat{x}_{i,t+1|t}$ (Prior)- User State Estimate of User i at time $t+1$ given all observations till time t (inclusive)
- ▶ $\hat{x}_{i,t|t}$ (Posterior) - User state Estimate of User i at time t given all observations till time t (inclusive)
- ▶ Same for P matrix too.
- ▶ $H_{i,t}$ is the subset of rows from V for which the observations $y_{i,t}$ is available or observed (therefore, $H_{i,t}$ is much smaller compared to V)

Extension to KF in EM - E Step

Forward Pass - Kalman Filter

$$\begin{aligned}\hat{x}_{i,t+1|t} &= A_{i,t+1}\hat{x}_{i,t|t} \\ P_{i,t+1|t} &= A_{i,t+1}P_{i,t|t}A_{i,t+1}^T + Q_{i,t} \\ K_{i,t} &= P_{i,t|t-1}H_{i,t}(H_{i,t}P_{i,t|t-1}H_{i,t}^T + R_{i,t})^{-1} \\ \hat{x}_{i,t|t} &= \hat{x}_{i,t|t-1} + K_{i,t}(y_{i,t} - H_{i,t}\hat{x}_{i,t|t-1}) \\ P_{i,t|t} &= P_{i,t|t-1} - K_{i,t}H_{i,t}P_{i,t|t-1}\end{aligned}\tag{2}$$

Backward Pass - RTS Smoothing

$$\begin{aligned}J_{i,t} &= P_{i,t|t}A_{i,t+1}^TP_{i,t|t-1}^{-1} \\ \hat{x}_{i,t|T} &= \hat{x}_{i,t|t} + J_{i,t}(\hat{x}_{i,t+1|T} - \hat{x}_{i,t+1|t}) \\ P_{i,t|T} &= P_{i,t|t} + J_{i,t}(P_{i,t+1|T} - P_{i,t+1|t})J_{i,t}^T \\ P_{i,T,T-1|T} &= (I - K_{i,T}H_{i,T})A_{i,T}P_{i,T-1|T-1} \\ P_{i,t,t-1|T} &= P_{i,t|t}J_{i,t-1}^T + J_{i,t}(P_{i,t+1,t|T} - A_{i,t+1}P_{i,t|t})J_{i,t-1}^T\end{aligned}\tag{3}$$

Extension to KF in EM - M Step

$$\begin{aligned}\log L &= \log(p(x, y; \theta)) \\ &= \sum_{i=1}^N \log(p(x_{i,0})) + \sum_{i=1}^N \sum_{t=1}^T \log(p(x_{i,t}|x_{i,t-1})) + \sum_{i=1}^N \sum_{t=1}^T \log(p(y_{i,t}|x_{i,t}))\end{aligned}$$

where

$$p(x_{i,0}) \sim N(x_{i,0}; \mu_i, \Sigma_i)$$

$$p(x_{i,t}|x_{i,t-1}) \sim N(x_{i,t}; A_{i,t}x_{i,t-1}, Q_i)$$

$$p(y_{i,t}|x_{i,t}) \sim N(y_{i,t}; H_{i,t}x_{i,t}, R_i)$$

(4)

After the M-step, we obtain estimates for $A, V, \hat{\sigma}_U^2, \hat{\sigma}_Q^2, \hat{\sigma}_R^2$. The estimates are obtained from the smoothed $x_{i,\hat{t}|T}$, $P_{i,\hat{t}|T}$ and P lag.

Assumptions

Simplifying Assumptions

- ▶ Movie Factor do not change much with time and hence estimated to be \hat{V} .
- ▶ All the $A_{i,t}$ are same (i.e. \hat{A}).
- ▶ All the process noise variances ($Q_{i,t}$) and measurement noise variances ($R_{i,t}$) are same and equal to $\hat{\sigma}_Q^2$ and $\hat{\sigma}_R^2$ respectively.
- ▶ All the initial user states are initialized from a Gaussian with zero-mean and variance $\hat{\sigma}_U^2$

Extension to KF in EM - E Step - after applying Assumptions

Forward Pass - Kalman Filter

$$\begin{aligned}\hat{x}_{i,t+1|t} &= A\hat{x}_{i,t|t} \\ P_{i,t+1|t} &= AP_{i,t|t}A^T + Q \\ K_{i,t} &= P_{i,t|t-1}H(HP_{i,t|t-1}H^T + R)^{-1} \\ \hat{x}_{i,t|t} &= \hat{x}_{i,t|t-1} + K_{i,t}(y_{i,t} - H\hat{x}_{i,t|t-1}) \\ P_{i,t|t} &= P_{i,t|t-1} - K_{i,t}HP_{i,t|t-1}\end{aligned}\tag{5}$$

Backward Pass - RTS Smoothing

$$\begin{aligned}J_{i,t} &= P_{i,t|t}A^T P_{i,t|t-1}^{-1} \\ \hat{x}_{i,t|T} &= \hat{x}_{i,t|t} + J_{i,t}(\hat{x}_{i,t+1|T} - \hat{x}_{i,t+1|t}) \\ P_{i,t|T} &= P_{i,t|t} + J_{i,t}(P_{i,t+1|T} - P_{i,t+1|t})J_{i,t}^T \\ P_{i,T,T-1|T} &= (I - K_{i,T}H)AP_{i,T-1|T-1} \\ P_{i,t,T-1|T} &= P_{i,t|t}J_{i,t-1}^T + J_{i,t}(P_{i,t+1,T-1|T} - AP_{i,t|t})J_{i,t-1}^T\end{aligned}\tag{6}$$

Problems we faced

- ▶ Memory - It should be noticed for one iteration of the E-Step, we require to store all the $\hat{x}_{i,t+1|t}$, $\hat{x}_{i,t|t}$, $P_{i,t+1|t}$ and $P_{i,t|t}$ for all users $i=1$ to N , $t=1$ to T .
- ▶ Cold-Start Problem
- ▶ Clustering not possible as there were many missing values

Applied Solutions to those problems

- ▶ Ensemble Kalman Filtering and Smoothing Extension to avoid storing the covariance matrices $P_{i,t+1|t}$ and $P_{i,t|t}$ but computed as and when needed. [PLH05] [SSL⁺10]
- ▶ User clustering using SVD and apply the entire KF in EM individually.

KF to Ensemble KF

Forward Pass - Kalman Filter

$$\begin{aligned}\hat{x}_{i,t+1|t} &= A_{i,t+1}\hat{x}_{i,t|t} \\ P_{i,t+1|t} &= A_{i,t+1}P_{i,t|t}A_{i,t+1}^T + Q_{i,t} \\ K_{i,t} &= P_{i,t|t-1}H_{i,t}(H_{i,t}P_{i,t|t-1}H_{i,t}^T + R_{i,t})^{-1} \\ \hat{x}_{i,t|t} &= \hat{x}_{i,t|t-1} + K_{i,t}(y_{i,t} - H_{i,t}\hat{x}_{i,t|t-1}) \\ P_{i,t|t} &= P_{i,t|t-1} - K_{i,t}H_{i,t}P_{i,t|t-1}\end{aligned}\tag{7}$$

Backward Pass - RTS Smoothing

$$\begin{aligned}J_{i,t} &= P_{i,t|t}A_{i,t+1}^T P_{i,t+1|t}^{-1} \\ \hat{x}_{i,t|T} &= \hat{x}_{i,t|t} + J_{i,t}(\hat{x}_{i,t+1|T} - \hat{x}_{i,t+1|t}) \\ P_{i,t|T} &= P_{i,t|t} + J_{i,t}(P_{i,t+1|T} - P_{i,t+1|t})J_{i,t}^T \\ P_{i,T,T-1|T} &= (I - K_{i,T}H_{i,T})A_{i,T}P_{i,T-1|T-1} \\ P_{i,t,t-1|T} &= P_{i,t|t}J_{i,t-1}^T + J_{i,t}(P_{i,t+1,t|T} - A_{i,t+1}P_{i,t|t})J_{i,t-1}^T\end{aligned}\tag{8}$$

KF to Ensemble KF

Forward Pass - Ensemble Kalman Filter

$$\hat{x}_{i,t+1|t} = A_{i,t+1}\hat{x}_{i,t|t} + q_{i,t} \quad q_{i,t} \sim N(0, Q_{i,t})$$

$$PH^T = \frac{1}{N_s - 1} \sum_{j=1}^{N_s} (x_{i,t+1|t}^j - \bar{x}_{i,t+1|t})(Hx_{i,t+1|t}^j - \overline{Hx_{i,t+1|t}})^T$$

$$HPH^T = \frac{1}{N_s - 1} \sum_{j=1}^{N_s} (Hx_{i,t+1|t}^j - \overline{Hx_{i,t+1|t}})(Hx_{i,t+1|t}^j - \overline{Hx_{i,t+1|t}})^T$$

$$K_{i,t} = P_{i,t|t-1}H_{i,t}(H_{i,t}P_{i,t|t-1}H_{i,t}^T + R_{i,t})^{-1}$$

$$\hat{x}_{i,t|t} = \hat{x}_{i,t|t-1} + K_{i,t}(y_{i,t} - H_{i,t}\hat{x}_{i,t|t-1}) \quad (9)$$

Backward Pass - Ensemble RTS Smoothing

$$\begin{aligned} P_{i,t|t-1}w &= (\hat{x}_{i,t+1|T} - \hat{x}_{i,t+1|t}) \\ \hat{x}_{i,t|T} &= \hat{x}_{i,t|t} + P_{i,t|t}Aw \end{aligned} \quad (10)$$

KF to Ensemble KF

Backward Pass - Ensemble RTS Smoothing (continued...)

$$\begin{aligned} P_{i,t|t} &= \frac{1}{N_s} \sum_{j=1}^{N_s} (x_{i,t|t}^j - \bar{x}_{i,t|t})(x_{i,t|t}^j - \bar{x}_{i,t|t})^T \\ P_{i,t|t-1} &= \frac{1}{N_s} \sum_{j=1}^{N_s} (x_{i,t|t-1}^j - \bar{x}_{i,t|t-1})(x_{i,t|t-1}^j - \bar{x}_{i,t|t-1})^T \end{aligned} \tag{11}$$

Results on KF,KS,EnKF,EnKS

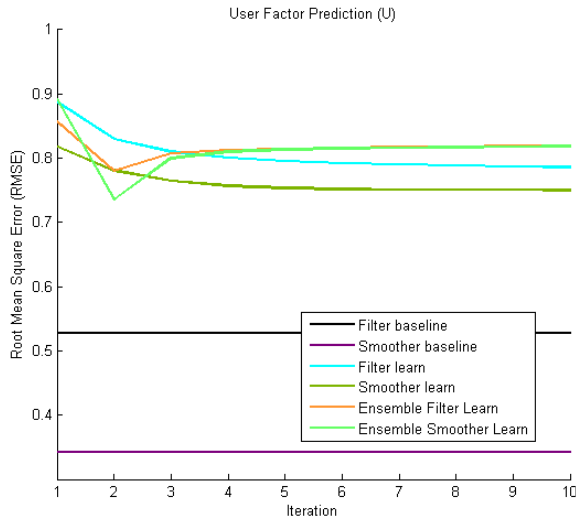


Figure 3 : UserFactor Prediction \hat{x} - $(N,M,T,K,E) = (300,200,20,20,10)$

Results on KF,KS,EnKF,EnKS

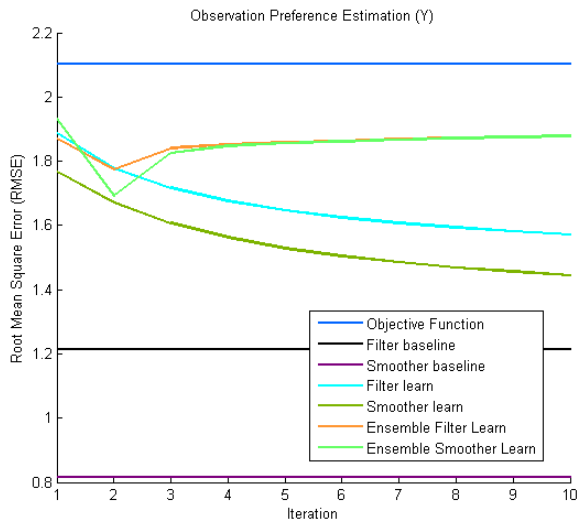


Figure 4 : ObsFactor Prediction \hat{y} - $(N,M,T,K,E) = (300,200,20,20,10)$

Results on KF,KS,EnKF,EnKS

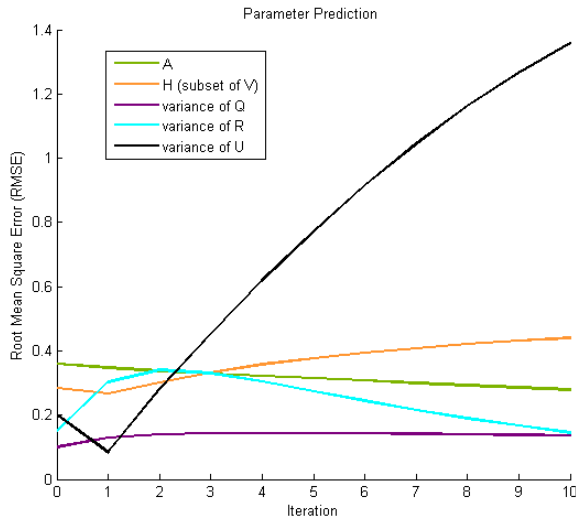


Figure 5 : Parameter Prediction - $(N,M,T,K,E) = (300,200,20,20,10)$

Results on Clustering and then doing KF,KS

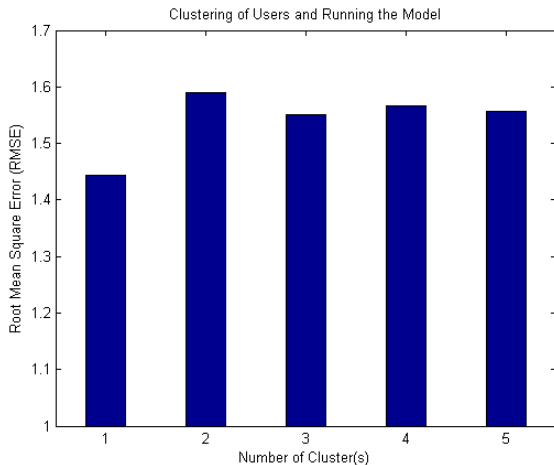


Figure 6 : Initial Clustering of Users and then running the model on each cluster(s)

Comparisons of All Extensions

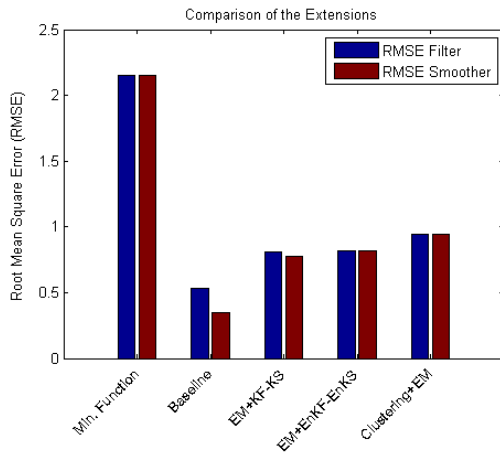







Figure 7 : Comparisons of All Extensions

Tabular Comparison including Run Times





Method Run	Time Taken (seconds)	RMSE (Filter)	RMSE (Smoother)
Stochastic Gradient Descent	165.2428	2.15	2.15
KF Baseline and KF Smoother Baseline	0.285073	0.53	0.345
KF and KS	5.041342	0.81	0.773
EnKF and EnKS	46.705548	0.82	0.82
Clustering of Users (3 clusters)	246.705548	0.94	0.94

Figure 8 : Tabular Comparison including Run Times

References I

-  Tristan Fletcher, *The kalman filter explained*, 2010.
-  Samuel Nowakowski, Cédric Bernier, and Anne Boyer, *A new recommender system based on target tracking: a kalman filter approach*, CoRR **abs/1012.3280** (2010).
-  HERSCHEL L. MITCHELL P. L. HOUTEKAMER, *Ensemble kalman filtering*, 2005.
-  Ruslan Salakhutdinov and Andriy Mnih, *Bayesian probabilistic matrix factorization using markov chain monte carlo*, In ICML '08: Proceedings of the 25th International Conference on Machine Learning, 2008.
-  "Ruslan Salakhutdinov and Andriy Mnih", "*probabilistic matrix factorization*", "Advances in Neural Information Processing Systems", vol. "20", "2008".

References II

-  Jonathan R. Stroud, Michael L. Stein, Barry M. Lesht, David J. Schwab, and Dmitry Beletsky, *An ensemble kalman filter and smoother for satellite data assimilation*, Journal of the American Statistical Association **105** (2010), no. 491, 978–990.
-  Nachiketa Sahoo, Param Vir Singh, and Tridas Mukhopadhyay, *A hidden markov model for collaborative filtering*, MIS Q. **36** (2012), no. 4, 1329–1356.
-  John Z. Sun, Kush R. Varshney, and Karthik Subbian, *Dynamic matrix factorization: A state space approach*, CoRR **abs/1110.2098** (2011).
-  Ariel Zetlin-Jones Ted Rosenbaum, *The kalman filter and the em algorithm*, 2006.

Acknowledgement

- ▶ Dr.Aleksandr Aravkin (IBM TJ Watson)
- ▶ Dr.John Sun (MIT, Research Lab of Electronics)
- ▶ Dr.Kush R. Varshney (MIT, IBM TJ Watson)
- ▶ Hyungtae Kim
- ▶ Guifan Li

Thank You ! Questions ?