Probabilistic Matrix Factorization

이종복

2022.01.04

contents

contents

Background

Explicit Feedback

Explicit Feedback
- Ratings
- Likes
- Written reviews

Implicit Feedback
- Click
- Listening pattern
- Browsing behaviors

Background – Matrix factorization

Collaborative filtering
: Information을 통한 관심사 자동 예측

Matrix factorization

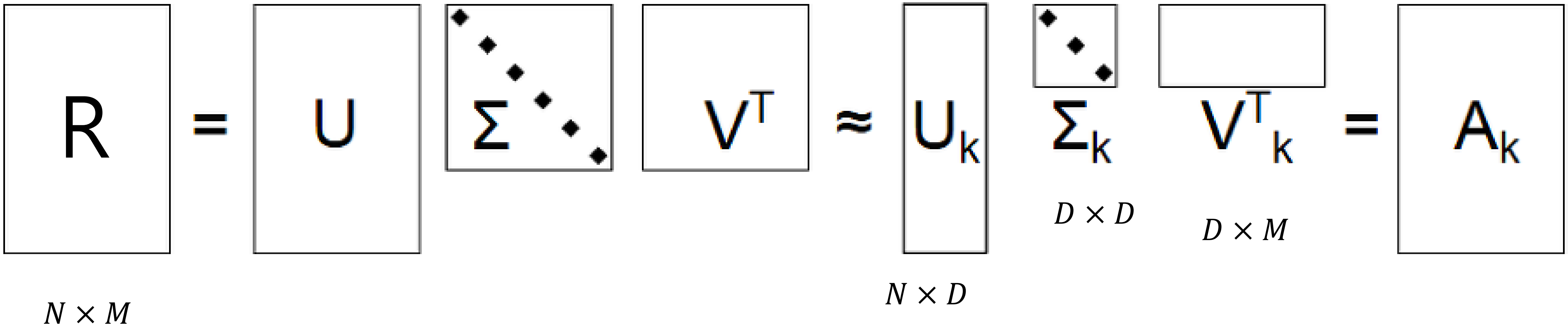
$$R = U^T V$$

$$U \rightarrow D\ latent\ features \times N\ users$$
$$V \rightarrow D\ latent\ features \times M\ movies$$
$$R \rightarrow N\ users \times M\ movies$$

U와 V로 target matrix R 예측. (under loss function, etc…)
* D -> 장르, 감독, 배우 등의 잠재적인 요소들

Background - SVD

- SVD(Singular Value Decomposition)



Background - SVD

- Low-rank approximations using SVD

Diagonal matrix Σ – latent feature의 strength

Ex) 영화 평가 Sum-Squared distance를 최소화하는 R, U, V

	스타워즈	아바타	스파이더맨	어벤져스	타이타닉
u1	1	1	1	0	0
u2	3	3	3	0	0
u3	0	0	0	5	5
u4	0	0	0	2	2
u5	4	?	3	?	2

u5의 평균인 3으로 대체

9.39	0.00	0.00	0.0	0.0
0.00	6.81	0.00	0.0	0.0
0.00	0.00	0.71	0.0	0.0
0.00	0.00	0.00	0.0	0.0
0.00	0.00	0.00	0.0	0.0

Σ

9.39	0.00	0.0	0.0	0.0
0.00	6.81	0.0	0.0	0.0
0.00	0.00	0.0	0.0	0.0
0.00	0.00	0.0	0.0	0.0
0.00	0.00	0.0	0.0	0.0

```
from numpy.linalg import svd
```

	스타워즈	아바타	스파이더맨	어벤져스	타이타닉
u1	1.087382	0.944252	0.944252	0.068472	-0.074658
u2	3.262146	2.832757	2.832757	0.205416	-0.223974
u3	0.112082	-0.071506	-0.071506	5.087827	4.904239
u4	0.044833	-0.028602	-0.028602	2.035131	1.961695
u5	3.742197	3.164472	3.164472	2.797988	2.220263

contents

Introduction

SVD의 한계

실제 Netflix 등에서는 user가 모든 영화를 보고 평가하지는 않기에 sparse한 matrix 형성
이는 minor modification에도 non – convex optimization problem이 됨.

이를 해결하기 위해 Low-rank approximation이 아닌 U, V의 norms를 penalizing하는 방법 또한 있으나, 이 또한 큰 dataset(over millions of observations)에서는 잘 작동하지 않음.

Introduction

In sparse and imbalanced datasets

나온 collaborative filtering algorithms의 문제점

1. Large scale dataset에 적합하지 않음
2. Few ratings에 대한 추론이 어려움.

실제 data는 위에서 제시된 대로 imbalance하기에 이러한 특성을 반영하는 collaborative filtering algorithm이 필요함.

-> PMF 제시

Introduction

Goals

1. Suggest PMF that have linear complexity, can respond imbalance datasets
2. Automatic Complexity Control for PMF Models
3. Suggest constraint PMF
4. Compare with SVD

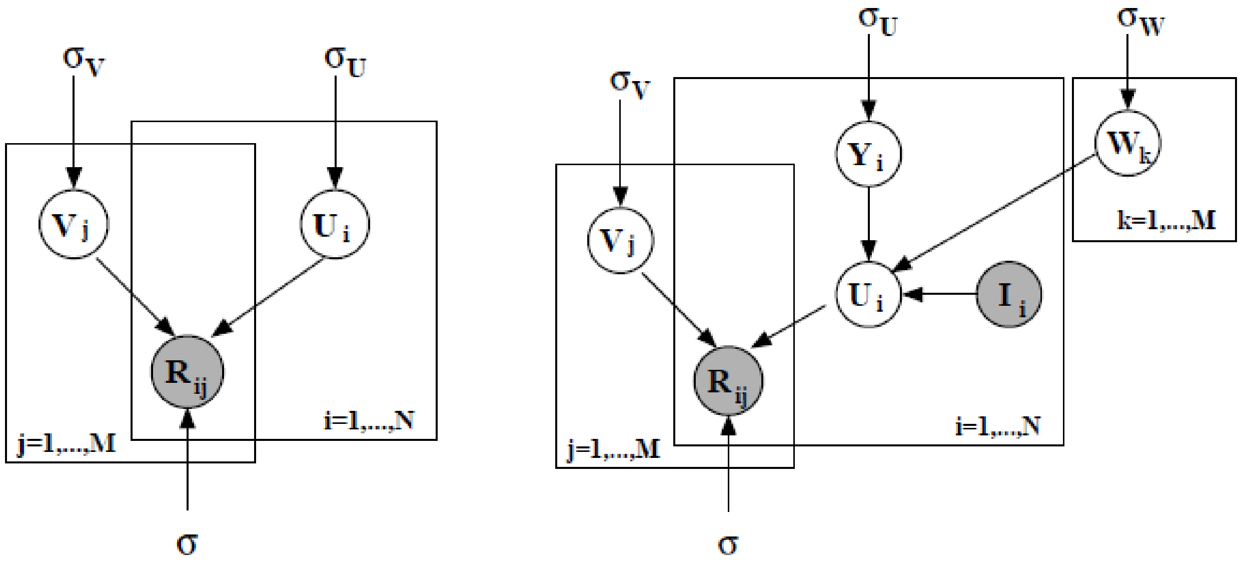


Figure 1: The left panel shows the graphical model for Probabilistic Matrix Factorization (PMF). The right panel shows the graphical model for constrained PMF.

contents

PMF

Prior Knowledge

Bayesian inference

사전 확률
(prior)

$$\boxed{P(H|E)} = \frac{P(E|H)\boxed{P(H)}}{P(E)}$$

사후 확률
(posterior)

MAP(Maximum A Posteriori) Estimation

$$\begin{aligned}\hat{\theta}_{\mathrm{MAP}}(x) &= \underset{\theta}{\arg\max}\ f(\theta \mid x) \\ &= \underset{\theta}{\arg\max}\ \frac{f(x \mid \theta)\, g(\theta)}{\int_{\Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta} \\ &= \underset{\theta}{\arg\max}\ f(x \mid \theta)\, g(\theta).\end{aligned}$$

Matrix factorization

$$R = U^T V$$

$$\begin{aligned} U &\rightarrow D\ latent\ features \times N\ users \\ V &\rightarrow D\ latent\ features \times M\ movies \\ R &\rightarrow N\ users\ \times M\ movies \end{aligned}$$

Use MAP

$$\underset{\mathrm{U,V}}{\operatorname{argmax}}\ p(U, V|R)$$

PMF

Assumption

1. User's evaluation follows the Gaussian distribution.

2. User의 선호도는 과거의 경향을 따라간다. (Collaborative Filtering의 기본 assumption)

3. Users who have rated similar sets of movies are likely to have similar preferences
   (section 4: Constrained PMF 에서 사용)

PMF

Notation

N users

M movies

U = D * N (latent user feature matrix) U_i: user i의 feature vector

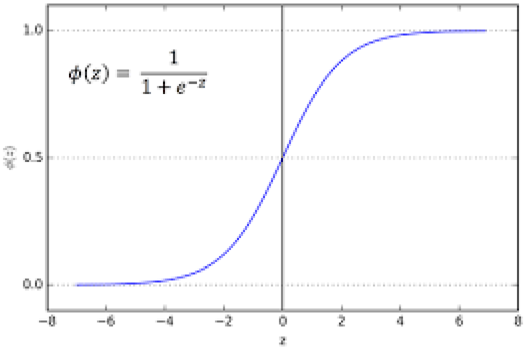
V = D * M (latent movie feature matrix) V_j: movie j의 feature vector

I_{ij}: user i가 movie j를 평가했으면 1, 평가하지 않았으면 0. indicator function.

$R = U^T V$: preference matrix

Rating: 1 to K rating (t(x) = (x-1)/(K-1) 로 [0,1]사이 값으로 변환)

g(x): sigmoid function ($\mathbb{R} \rightarrow [0,1]$)



PMF

Likelihood, Prior, Log Posterior

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)\right]^{I_{ij}}, \qquad (1)$$

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}). \qquad (2)$$

Instead of $U^T V$,
we use $g(U^T V)$

Also, we map ratings
[1, K] to the interval
[0, 1] using the t(x)

MAP

Maximize
Log posterior

$$\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j$$
$$-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\right)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\right) + C, \qquad (3)$$

PMF

MAP, minimizing the sum-of-squared-errors

MAP

Maximize
Log posterior

$$\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-U_i^T V_j)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j$$
$$-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\right)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\right) + C, \qquad (3)$$

=

Minimizing
E

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij}-U_i^T V_j\right)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \| V_j \|_{Fro}^2, \qquad (4)$$

E = (sum of squared errors objective function) + Frobenius regularization terms

$\lambda_U = \frac{\sigma^2}{{\sigma_U}^2}, \lambda_V = \frac{\sigma^2}{{\sigma_V}^2}$

모든 rating이 관측된다면, SVD와 동일

-> ${\sigma_U}^2$, ${\sigma_V}^2$ goes infinity -> λ_U, λ_V goes zero

contents

Automatic Complexity Control for PMF Models

Make well-generalize PMF model

Capacity control is essential

Simple methods

1. Reduce dimensionality of feature vector
2. Regularized by penalizing term

이러한 방법은 앞에서 보았듯 dataset이 imbalanced한 경우(일부 유저가 높은 평가율을 보이는 경우) 효과적이지 않고, 계산 비용이 비싸다는 단점이 있음.

논문에서는 parameter, hyperparameter를 자동적으로 찾아내는 방법을 제안함.

Model의 complexity는 hyperparameter에 의해 제어되기 때문.

Automatic Complexity Control for PMF Models

Summary

Point estimate of parameters and hyperparameters by maximizing log-posterior

$$\ln p(U, V, \sigma^2, \Theta_U, \Theta_V | R) = \ln p(R|U, V, \sigma^2) + \ln p(U|\Theta_U) + \ln p(V|\Theta_V) + \ln p(\Theta_U) + \ln p(\Theta_V) + C, \quad (6)$$

where Θ_U and Θ_V are the hyperparameters for the priors over user and movie feature vectors respectively and C is a constant that does not depend on the parameters or hyperparameters.

Method

If prior is Gaussian -> gradient ascent사용하여 feature vector를 업데이트(closed form sol)

If prior is Mixture of Gaussian-> EM을 사용하여 업데이트.

contents

Constrained PMF

Latent similarity constraint matrix W

Let $W \in R^{D \times M}$ be a latent similarity constraint matrix.
직관적으로, W_k는 k번째 영화의 영향력이라 할 수 있음.
Prior mean인 0에서 W_k에 의해 이동.
Y_i는 offset. U_i는 기존의 distribution에서 W_k에 의하여 이동.

$$U_i = Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}}. \tag{7}$$

Constrained PMF

Latent similarity constraint matrix W

Assumption 3: (Slide 11)

Users who have rated similar sets of movies are likely to have similar preferences

W는 이전에 그 영화를 봤던 사용자들의 평가가 미치는 영향

$$p(W|\sigma_W) = \prod_{k=1}^{M} \mathcal{N}(W_k|0, \sigma_W^2 \mathbf{I}).$$

$$U_i = Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}}.$$

Constrained PMF

Final result

Use GD to search Y, V, W -> can scales linearly(# of observation)

$$p(R|Y,V,W,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}\left(R_{ij}|g\left(\left[Y_i + \frac{\sum_{k=1}^{M} I_{ik}W_k}{\sum_{k=1}^{M} I_{ik}}\right]^T V_j\right), \sigma^2\right)\right]^{I_{ij}}. \quad (8)$$

$$\begin{aligned} E \quad = \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\Big(R_{ij} - g\big(\big[Y_i + \frac{\sum_{k=1}^{M} I_{ik}W_k}{\sum_{k=1}^{M} I_{ik}}\big]^T V_j\big)\Big)^2 \quad (10) \\ & + \frac{\lambda_Y}{2}\sum_{i=1}^{N} \parallel Y_i \parallel_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \parallel V_j \parallel_{Fro}^2 + \frac{\lambda_W}{2}\sum_{k=1}^{M} \parallel W_k \parallel_{Fro}^2, \end{aligned}$$

Constrained PMF

Summary

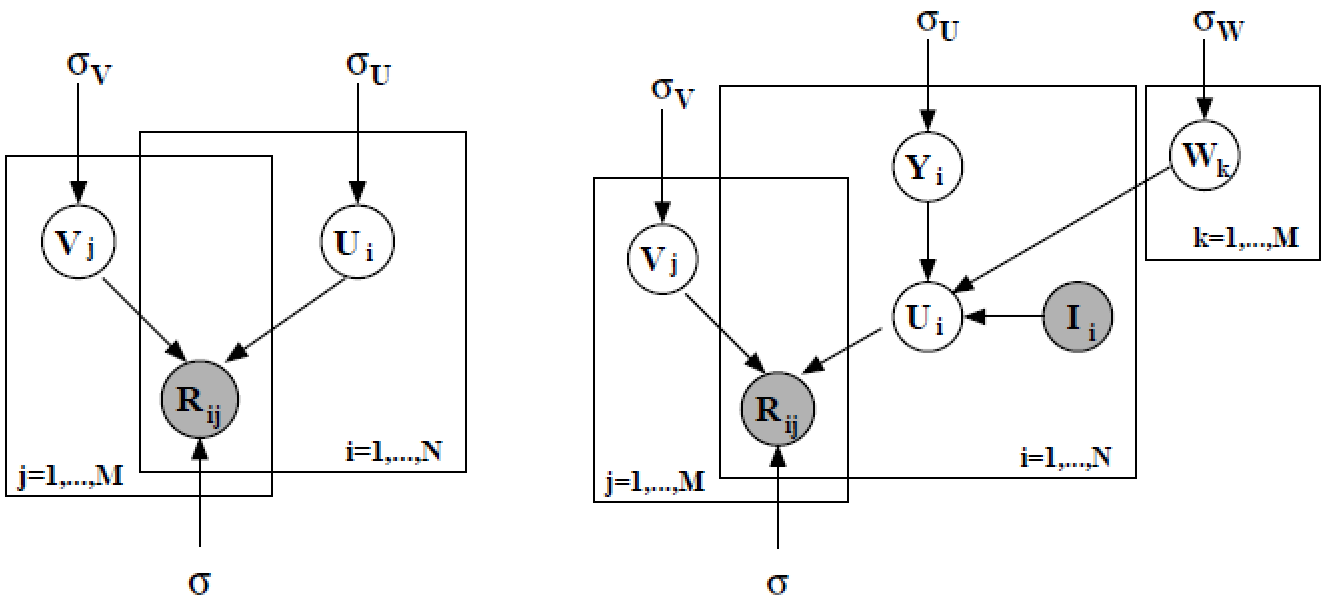


Figure 1: The left panel shows the graphical model for Probabilistic Matrix Factorization (PMF). The right panel shows the graphical model for constrained PMF.

Constrained PMF는 few rating을 가진 user에 효과적

contents

Experimental Results

1. Use Netflix Data, 100,000 mini-batch (5.1, 5.2)

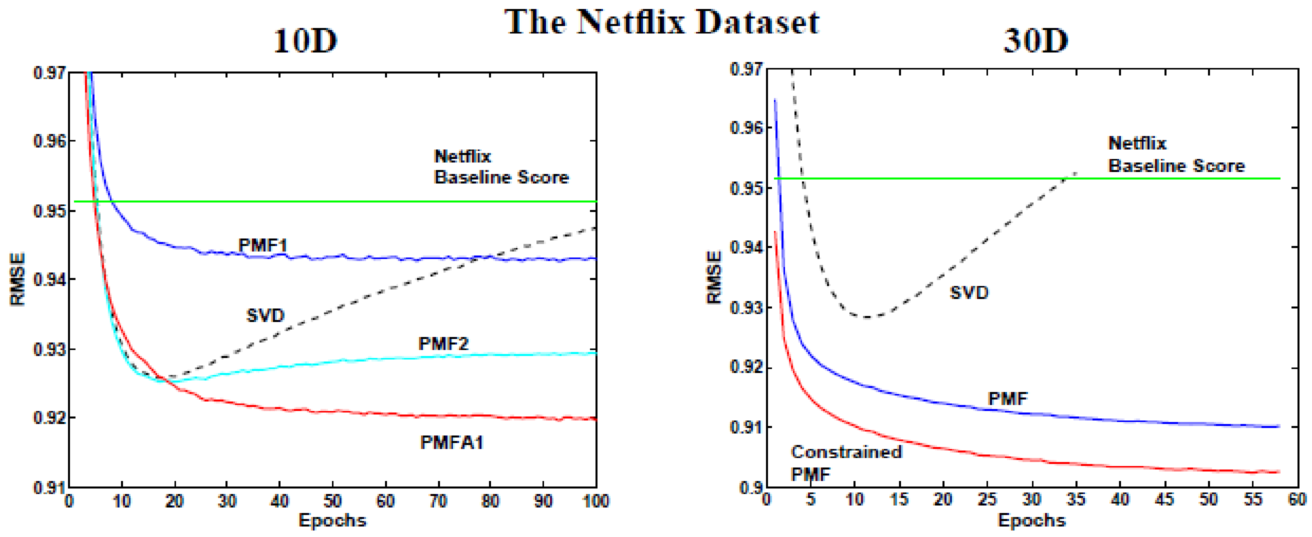
2. Evaluation PMF with adaptive prior (5.3)

PMF1: λ_U=0.01, λ_V=0.001

PMF2: λ_U=0.001, λ_V=0.0001

PMFA1: spherical covariance (adaptive)

PMFA2: diagonal covariance (adaptive)



SVD overfit, PMF1 underfit

-> adaptive한 경우가 real-world dataset에서 효과가 좋다.

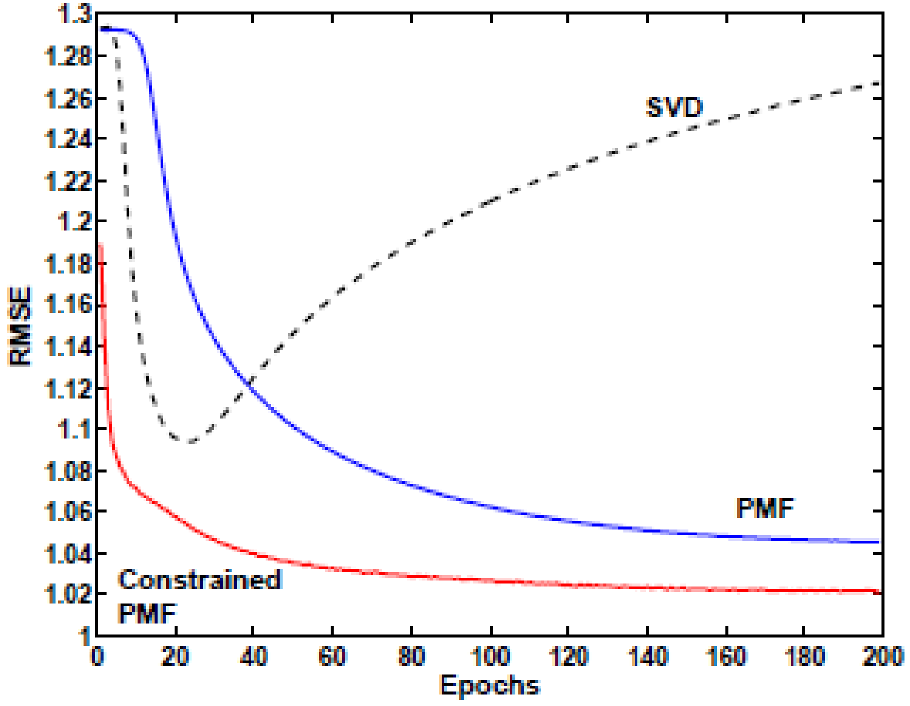
Experimental Results

3. Results for Constrained PMF (5.4)

1. SVD – overfit

2. Constrained PMF converges faster than unconstrained PMF

3. Observation이 많아질수록 Constrained와 unconstrained가 비슷.



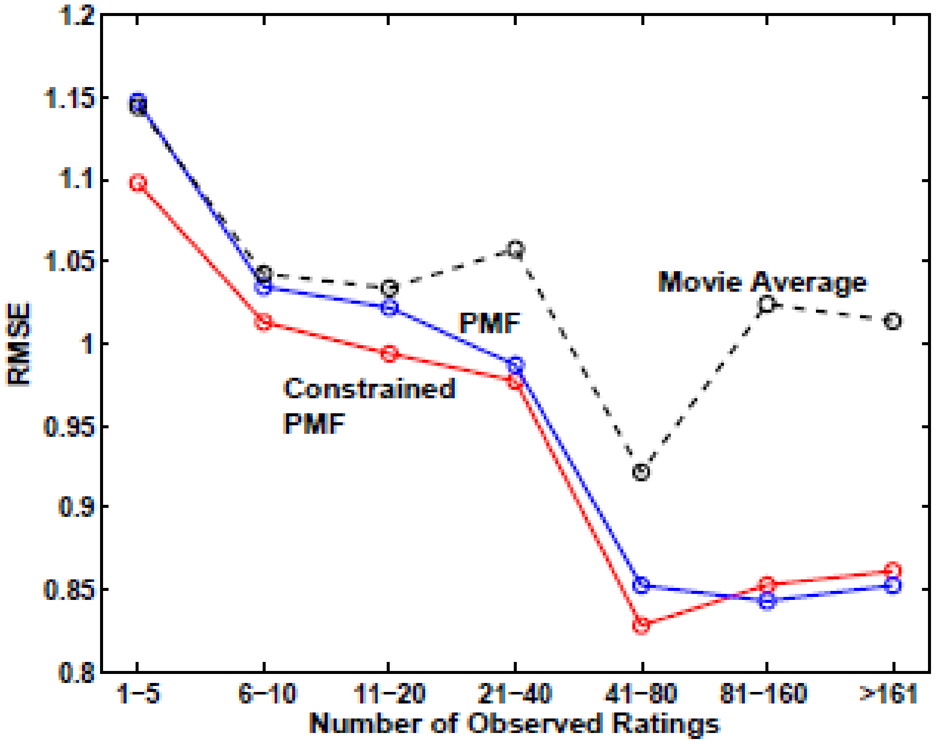
Experimental Results

4. Additional discussion (5.4)

1. rating을 모르는 경우에서 movie average보다 좋은 성능을 냄. 영화 시청 여부만 파악한다면, PMF를 통한 예측이 가능.

2. Whole Netflix dataset을 사용했을 때하고 비슷한 performance를 냄.

3. PMF, PMF adaptive, Constrained PMF를 선형적으로 조합하면 보다 낮은 error rate생성.



contents

- Background - SVD
- Introduction
- Probabilistic Matrix Factorization (PMF)
- Automatic Complexity Control for PMF Models
- Constrained PMF
- Experimental Results
- Summary and Discussion

Summary and Discussion

Summary (6)

1. PMF 개념 설명

2. PMF with adaptive prior (Automatic Complexity Control for PMF Models)

3. Constrained PMF

4. Result

Summary and Discussion

Discussion (6, additional)

1. Automatic Complexity Control에서 hyperparameter를 point estimates를 이용하여 찾음. 만약 full posterior distribution (fully Bayesian method) 을 찾는다면 MCMC(Markov chain Monte Carlo)를 이용해 찾을 수 있음. ->computationally more expensive, but lead significant effect

2. Depend on assumption

3. Task specific model – Matrix Factorization의 특징

4. Difficulty in interpreting feature vector

각 feature들의 의미가 직관적이지 않음 – Matrix Factorization의 특징

감사합니다

이종복

2022.01.04