

SoRec

본 논문에서 제안하는 SoRec 모델은 Probabilistic Matrix Factorization (PMF)에 Social Network Information을 추가하여 확장한 모델이다.

PMF vs SoRec

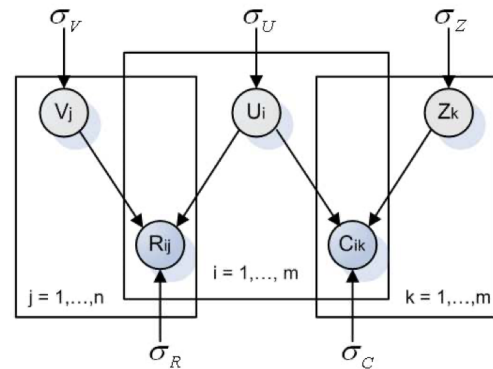
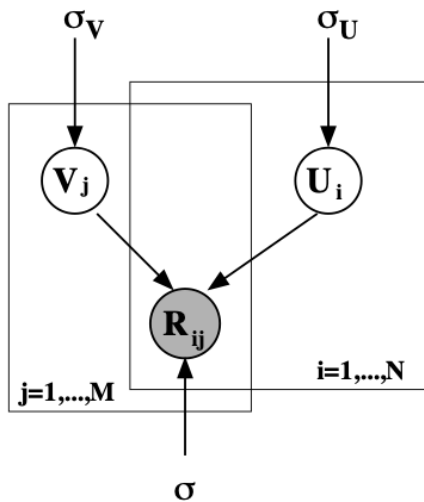


Figure 2: Graphical Model for Social Recommendation

Introduction

Collaborative Filtering은 “어떠한 유저가 있을 때, 그 유저와 비슷한 유저들이 좋아하는 아이템은 원래의 유저도 좋아할 것이다”라는 가정에서 시작한다. 이러한 방법은 넷플릭스, 아마존 같은 플랫폼에서 성공을 거뒀지만 아래와 같은 문제점들이 존재한다.

1. Sparsity : density of available ratings is less than 1%
2. Cold Start Problem: cannot handle users who have never rated any items
3. i.i.d assumption: 현실에서는 user들이 모두 독립적인 게 아니라 아이템을 선택하기 전 의견을 물어보곤 함

이러한 문제들을 해결하기 위해 Social Network information과 user-item rating matrix를 같이 고려하는 Social Recommendation 모델을 제안한다. 이 방식을 통해

large datasets에도 데이터셋의 크기에 맞게 linear하게 증가함을 보여 efficient 하다고 주장한다.

Social Recommendation Framework

U_i : latent feature vector of user _{i}

Z_j : factor – specific vector of user _{j}

V_k : latent feature vector of item _{k}

Social Network Matrix Factorization

$$p(C|U, Z, \sigma_C^2) = \prod_{i=1}^m \prod_{k=1}^m \mathcal{N} \left[\left(c_{ik} | g(U_i^T Z_k), \sigma_C^2 \right) \right]^{I_{ik}^C}$$

- C : social network matrix
- U, Z, σ_C^2 가 주어졌을 때 C 라는 Matrix가 나타날 확률은 각 Component들의 Probability Product로 계산
- Probability는 $g(U_i^T Z_k)$ 를 mean으로, σ_C^2 를 std로 갖는 gaussian distribution의 pdf로 정의
- $g(x) = 1/(1 + e^{-x})$
- I_{ik}^C : Indicator function of C_{ik}

Prior of feature vectors

$$p(U|\sigma_U^2) = \prod_{i=1}^m \mathcal{N}(U_i | 0, \sigma_U^2 \mathbf{I}),$$

$$p(Z|\sigma_Z^2) = \prod_{k=1}^m \mathcal{N}(Z_k | 0, \sigma_Z^2 \mathbf{I}).$$

U, Z 는 zero-mean spherical Gaussian priors를 가짐

Maximum A Posteriori

과정

$$\hat{\theta}_{MAP} := \underset{\theta}{\operatorname{argmax}} p(\theta|D)$$

$$\begin{aligned}\hat{\theta}_{MAP} &:= \underset{\theta}{\operatorname{argmax}} p(\theta | \mathcal{D}) \\ &= \underset{\theta}{\operatorname{argmax}} \frac{p(\mathcal{D} | \theta)p(\theta)}{p(\mathcal{D})} \\ &= \underset{\theta}{\operatorname{argmax}} p(\mathcal{D} | \theta)p(\theta)\end{aligned}$$

MAP 적용

$$\begin{aligned}p(U, Z|C, \sigma_C^2, \sigma_U^2, \sigma_Z^2) \\ &\propto p(C|U, Z, \sigma_C^2)p(U|\sigma_U^2)p(Z|\sigma_Z^2) \\ &= \prod_{i=1}^m \prod_{k=1}^m \mathcal{N} \left[\left(c_{ik} | g(U_i^T Z_k), \sigma_C^2 \right) \right]^{I_{ik}^C} \\ &\times \prod_{i=1}^m \mathcal{N}(U_i | 0, \sigma_U^2 \mathbf{I}) \times \prod_{k=1}^m \mathcal{N}(Z_k | 0, \sigma_Z^2 \mathbf{I}).\end{aligned}$$

- 위와 같은 과정을 적용하면 아래와 같이 나타낼 수 있다.
- 하지만 C_{ik} 는 $user_i, user_k$ 의 특성에 따라 달라져야 보다 정확한 결과를 얻을 수 있다.
 - 예를 들어, $user_i$ 가 여러 사람을 follow하는 경우에는 C_{ik} 를 줄여야 할 것이고, 반면 $user_k$ 를 follow 하는 사람이 많다면 C_{ik} 를 크게 만들어야 할 것이다.
- 이러한 문제점을 해결하기 위해 새롭게 정의된 C_{ik}^* 는 아래와 같다.

$$p(C|U, Z, \sigma_C^2) = \prod_{i=1}^m \prod_{j=1}^n \mathcal{N} \left[\left(c_{ik}^* | g(U_i^T Z_k), \sigma_C^2 \right) \right]^{I_{ik}^C}$$

$$c_{ik}^* = \sqrt{\frac{d^-(v_k)}{d^+(v_i) + d^-(v_k)}} \times c_{ik},$$

User-item Matrix Factorization

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^m \prod_{j=1}^n \mathcal{N} \left[\left(r_{ij} | g(U_i^T V_j), \sigma_R^2 \right) \right]^{I_{ij}^R},$$

R : User-item Rating Matrix

Priors

$$p(U|\sigma_U^2) = \prod_{i=1}^m \mathcal{N}(U_i | 0, \sigma_U^2 \mathbf{I}),$$

$$p(V|\sigma_V^2) = \prod_{j=1}^n \mathcal{N}(V_j | 0, \sigma_V^2 \mathbf{I}).$$

MAP

$$\begin{aligned}
& p(U, V | R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \\
& \propto p(R | U, V, \sigma_R^2) p(U | \sigma_U^2) p(V | \sigma_V^2) \\
& = \prod_{i=1}^m \prod_{j=1}^n \mathcal{N} \left[\left(r_{ij} | g(U_i^T V_j), \sigma_R^2 \right) \right]^{I_{ij}^R} \\
& \times \prod_{i=1}^m \mathcal{N}(U_i | 0, \sigma_U^2 \mathbf{I}) \times \prod_{j=1}^n \mathcal{N}(V_j | 0, \sigma_V^2 \mathbf{I}).
\end{aligned}$$

- Social Network Matrix Factorization에서 진행했던 과정과 동일하게 진행

Matrix Factorization for Social Recommendation

$$\begin{aligned}
\ln p(U, V, Z | C, R, \sigma_C^2, \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_Z^2) = & \\
& -\frac{1}{2\sigma_R^2} \sum_{i=1}^m \sum_{j=1}^n I_{ij}^R (r_{ij} - g(U_i^T V_j))^2 \\
& -\frac{1}{2\sigma_C^2} \sum_{i=1}^m \sum_{k=1}^m I_{ik}^C (c_{ik}^* - g(U_i^T Z_k))^2 \\
& -\frac{1}{2\sigma_U^2} \sum_{i=1}^m U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^n V_j^T V_j - \frac{1}{2\sigma_Z^2} \sum_{k=1}^m Z_k^T Z_k \\
& -\frac{1}{2} \left(\left(\sum_{i=1}^m \sum_{j=1}^n I_{ij}^R \right) \ln \sigma_R^2 + \left(\sum_{i=1}^m \sum_{k=1}^m I_{ik}^C \right) \ln \sigma_C^2 \right) \\
& -\frac{1}{2} (m \ln \sigma_U^2 + n \ln \sigma_V^2 + m \ln \sigma_Z^2) + \mathcal{C}, \tag{8}
\end{aligned}$$

- posterior distribution에 log를 씌우면 위와 같이 정리된다

- maximizing the log-posterior는 아래 SSE objective function을 minimizing하는 것과 동일하다.

$$\begin{aligned} \mathcal{L}(R, C, U, V, Z) = & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij}^R (r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_C}{2} \sum_{i=1}^m \sum_{k=1}^m I_{ik}^C (c_{ik}^* - g(U_i^T Z_k))^2 \\ & + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 + \frac{\lambda_Z}{2} \|Z\|_F^2, \end{aligned} \quad (9)$$

Gradient Descent

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial U_i} &= \sum_{j=1}^n I_{ij}^R g'(U_i^T V_j) (g(U_i^T V_j) - r_{ij}) V_j \\ &\quad + \lambda_C \sum_{k=1}^m I_{ik}^C g'(U_i^T Z_k) (g(U_i^T Z_k) - c_{ik}^*) Z_k + \lambda_U U_i, \\ \frac{\partial \mathcal{L}}{\partial V_j} &= \sum_{i=1}^m I_{ij}^R g'(U_i^T V_j) (g(U_i^T V_j) - r_{ij}) U_i + \lambda_V V_j, \\ \frac{\partial \mathcal{L}}{\partial Z_k} &= \lambda_C \sum_{i=1}^m I_{ik}^C g'(U_i^T Z_k) (g(U_i^T Z_k) - c_{ik}^*) U_i + \lambda_Z Z_k, \end{aligned} \quad (10)$$

- 위의 Loss에서 학습 대상은 U, V, Z 기 때문에 이에 대한 Gradient를 나타내면 위와 같다.

Experimental analysis

위 저자는 아래 5가지 궁금증을 확인하기 위해 실험을 진행했다고 한다.

1. 제안한 모델이 SOTA 모델들과 비교했을 때 어떠한 성능을 갖는지
2. Social Information을 표현하는 정도를 뜻하는 λ_C 가 정확도에 어떠한 영향을 끼치는지
3. different observed ratings를 갖는 유저들의 performance는 어떻게 나타나는지
4. Cold Start 상황에서 어떠한 성능을 보이는지

5. large dataset에도 잘 적용될 수 있는지

Results

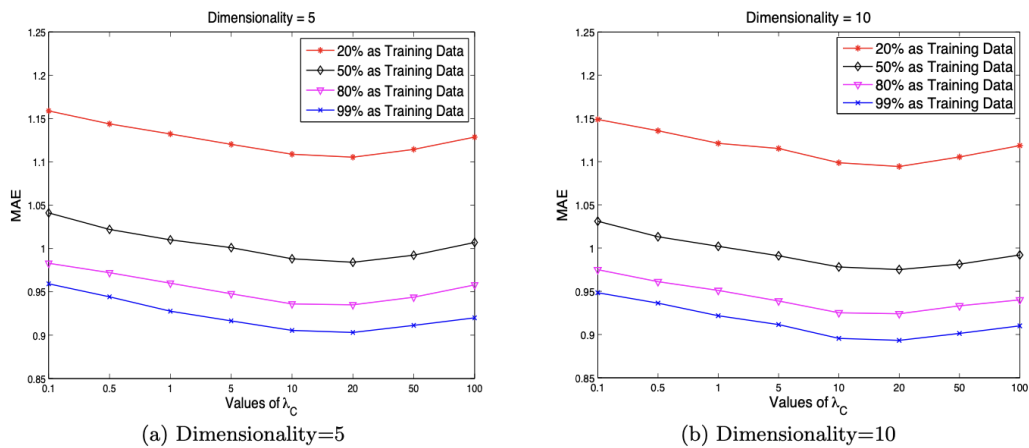
SoRec Vs SOTA

Table 2: MAE comparison with other approaches (A smaller MAE value means a better performance)

Training Data	Dimensionality = 5				Dimensionality = 10			
	MMMF	PMF	CPMF	SoRec	MMMF	PMF	CPMF	SoRec
99%	1.0008	0.9971	0.9842	0.9018	0.9916	0.9885	0.9746	0.8932
80%	1.0371	1.0277	0.9998	0.9321	1.0275	1.0182	0.9923	0.9240
50%	1.1147	1.0972	1.0747	0.9838	1.1012	1.0857	1.0632	0.9751
20%	1.2532	1.2397	1.1981	1.1069	1.2413	1.2276	1.1864	1.0944

- Metric으로 MAE를 사용함
- 제안한 모델이 MMMF, PMF, CPMF에 비해 좋은 성능을 나타내고 있음
 - 위의 모델 순대로 각각 7.82%, 9.98%, 11.01%의 정확도가 증가하여 상당히 증가 폭이 큰 것을 확인할 수 있음

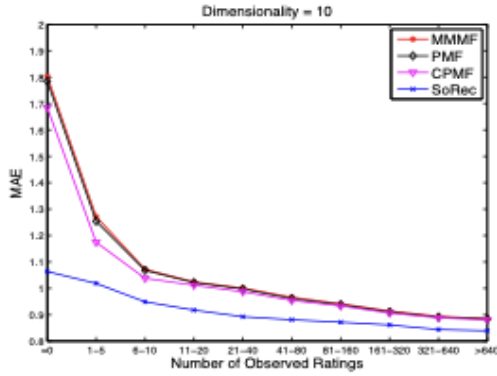
Impact of Parameter λ_C



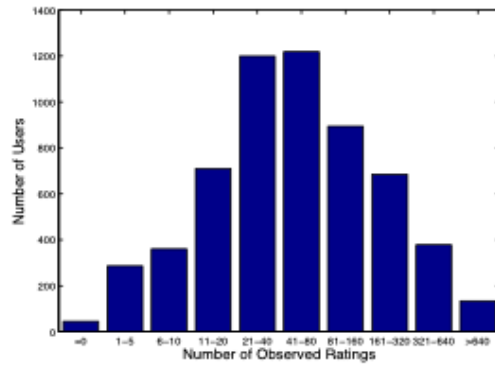
- λ_C 는 Social Information을 얼마나 반영할지 정하는 Hyperparameter
- 너무 크거나 작은 값은 오히려 Overfitting을 유발시켜 성능 저하가 발생함
- [10,20]의 값을 사용했을 때 Best Performance를 나타냄

Performance on Different Users

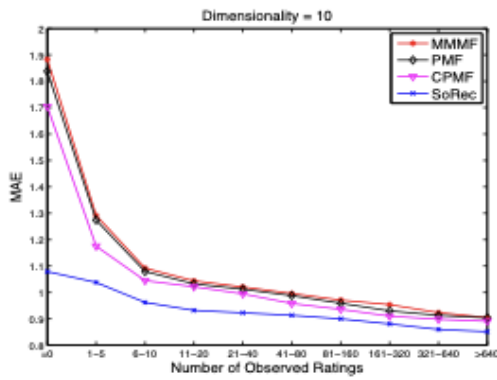
- User들의 # of rated에 따라 10개의 그룹으로 나타냄



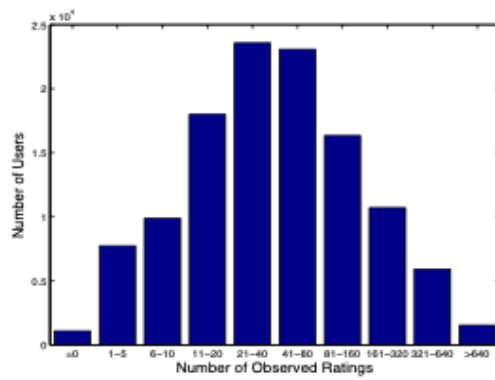
(a) Performance Comparison on Different User Rating Scales (99% as Training Data)



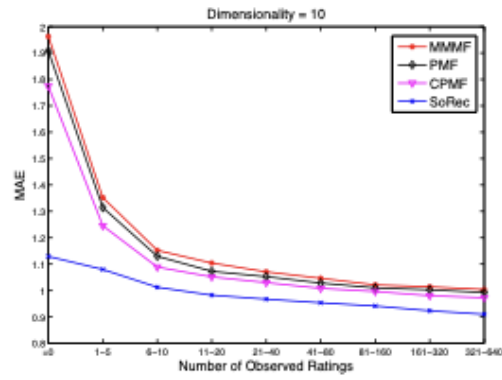
(b) Distribution of Testing Data (99% as Training Data)



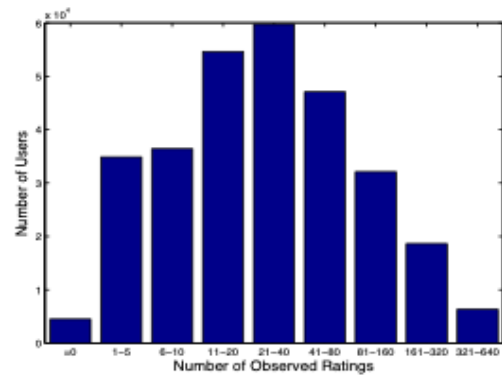
(c) Performance Comparison on Different User Rating Scales (80% as Training Data)



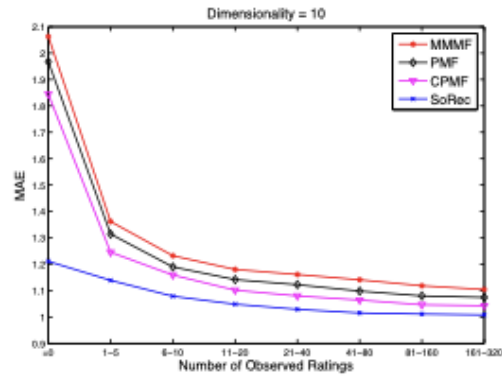
(d) Distribution of Testing Data (80% as Training Data)



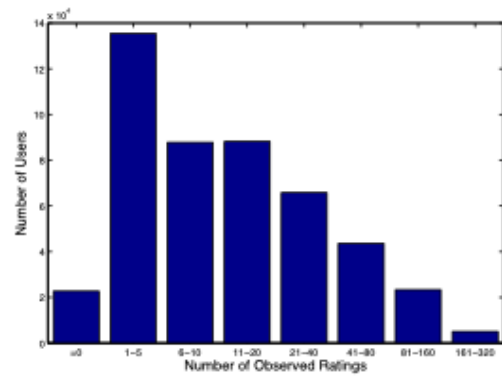
(e) Performance Comparison on Different User Rating Scales (50% as Training Data)



(f) Distribution of Testing Data (50% as Training Data)



(g) Performance Comparison on Different User Rating Scales (20% as Training Data)



(h) Distribution of Testing Data (20% as Training Data)

Figure 5: Performance Comparison on Different Users

- 이렇게 # of rated가 다른 그룹 사이에서도 가장 좋은 Performance를 나타내고 있음을 확인할 수 있음
- 특히, 적은 # of rated에서 좋은 성능을 나타냄을 확인할 수 있다. ("=0")

Efficiency Analysis

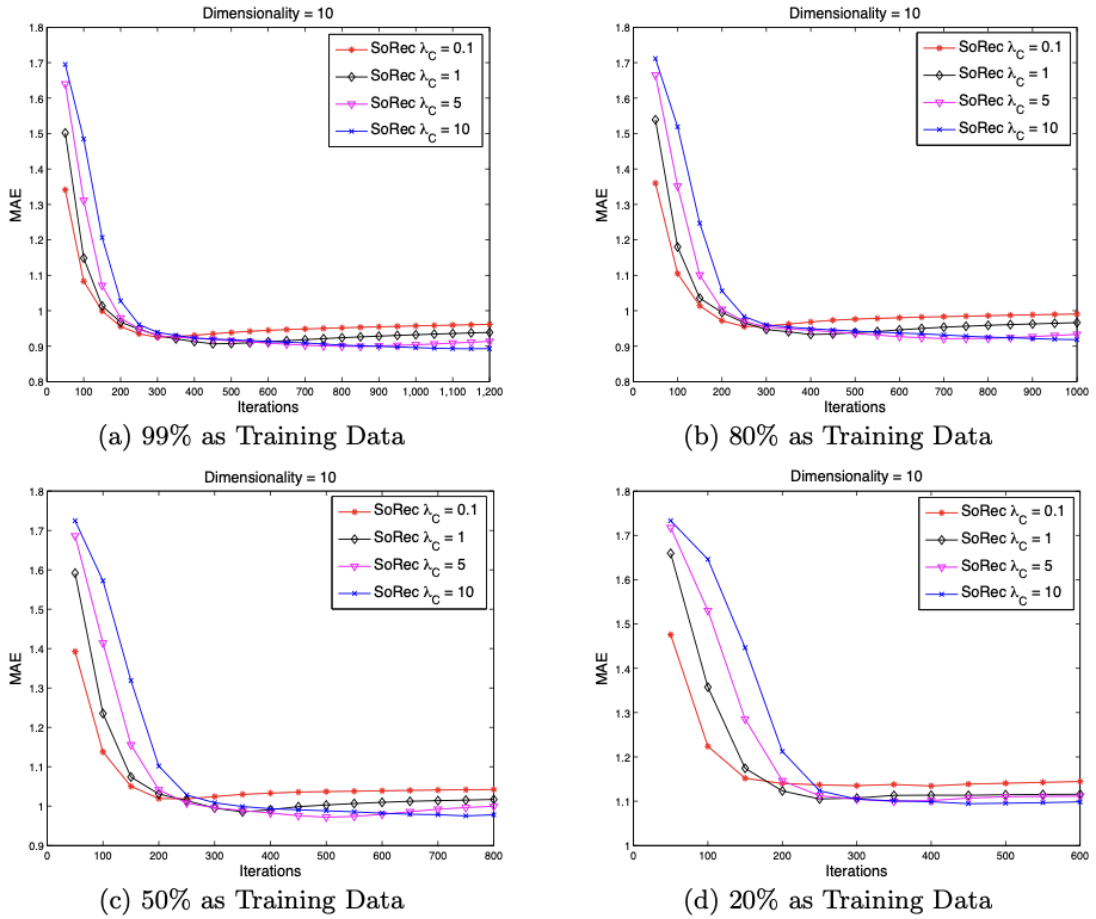


Figure 6: Efficiency Analysis

- Loss를 계산할 때 R, C 의 요소가 있는 곳인, 즉 indicator function이 1인 곳만을 이용하기 때문에 $O(\rho_R l + \rho_C l)$ 로 표현된다.

- Gradient 계산 시

- $\frac{\partial L}{\partial U} : O(\rho_R l + \rho_C l)$
- $\frac{\partial L}{\partial V} : O(\rho_R l)$
- $\frac{\partial L}{\partial Z} : O(\rho_C l)$

만큼 소요되기 때문에 1-iteration에서 소요되는 시간은 $O(\rho_R l + \rho_C l)$ 으로 dataset 크기에 linear하게 늘어나 효과적으로 large dataset에도 적용될 수 있다.

- 1-iteration에 걸리는 시간이 1초 미만으로 걸린다. 이를 바탕으로 99%를 Training-data로 사용한 Fig. 6(a)에서 나타난 Convergence에 걸리는 iteration이 1200을 이용하면 총 18분밖에 걸리지 않아 large dataset에도 적용할 수 있음을 보여준다.

Conclusion and Future Work

- user-item rating matrix와 social network를 결합한 novel social recommendation framework인 SoRec 모델이 제안되었음
- information retrieval / data mining과 같은 다른 task에도 적용될 수 있어 효용이 크다고 주장

Future Work

- 단순히 logistic function을 kernel로 사용하지 않고 Gaussian kernel 또는 polynomial kernel 과 같은 다른 kernel representation을 사용하면 성능을 증가시킬 수도 있을 것이라고 언급
- 또한 현재는 trust-information만 반영했는데 distrust information까지 반영하면 더 좋을 수도 ?
 - 단, 개인정보 문제가 있고 latent feature space에서 기존 벡터 표현과 다를 가능성 존재
- user들간의 information diffusion이 발생하는데 무시했기 때문에 이를 개선하면 좋을 듯