

# SoReg

본 저자는 Traditional Recommender System에 속하는 일반적인 Matrix Factorization, Neighborhood Model들은 Recommender System에서 좋은 Performance를 내고 있지만 User의 Social Relationship을 반영하지 않는 문제점이 존재한다고 주장한다.

예를 들어 우리는 단순히 우리의 판단만으로 상품을 구매하는 것이 아니라 이미 구매하거나 그 제품에 대해 알고 있는 친구, 지인들에게 의견을 묻고 구매한다. 이러한 친구, 지인, 가족들을 Social Relationship이라고 하는데, 이를 반영하면 Accuracy나 Personalized Recommendation을 개선할 수 있다고 제안한다.

즉, 이 Paper의 Novelty는 다음과 같이 요약할 수 있다.

1. 어떻게 Social Network Information이 Recommender System에 쓰일 수 있는지
2. social-based recommender system VS. trust-aware recommender system
3. Social Regularization: New matrix factorization objective with Social Network Information
4. 제안된 모델이 other contextual information을 어떻게 반영할 수 있는지 보여준다고 한다.

## Introduction

저자는 1) Traditional recommender system, 2) Trust-aware recommender system, 3) Social recommender system을 명확히 범주화하고 있기 때문에 이에 대한 이해가 선제적으로 필요하다.

### 1) Traditional recommender system

- Platform 안에서 발생하는 좋아요, 팔로우 같은 Trust 및 Social Information을 고려하는 것이 아니라 단순히 User들이 모두 동일하고 독립적인 관계 (i.i.d)를 갖는다고 생각하는 모델

### 2) Trust-aware recommender system

- Platform 안에서 Trust list에 추가하거나 좋아요, 팔로우와 같은 관계가 있다면 Trust relationship을 가진다.

- 이 방법은 User들은 그들이 Trust를 표현하는 상대들의 취향과 비슷할 것이라고 가정 하에 구성된다.
- 이러한 Trust는 Real-life에서 관계를 가지는 Social friendship과 다름을 구체적으로 명시한다.

### 3) Social recommender system

- Real-life에서 관계를 갖는 지인, 가족, 친구 등을 정보를 포함한 모델
- 이들의 취향은 매우 다양할 수 있기 때문에 효과적으로 반영하는 방식이 필요하다고 말한다.

## Problem Definition

본 모델은 User-item의 Rating matrix에서 Missing value를 Predict하는 문제를 해결하기 위한 목적으로 정의한다.

## Social Regularization

### Model 1: Average-based Regularization

$$\begin{aligned}
\min_{U,V} \mathcal{L}_1(R, U, V) &= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2 \\
&+ \frac{\alpha}{2} \sum_{i=1}^m \|U_i - \frac{1}{|\mathcal{F}^+(i)|} \sum_{f \in \mathcal{F}^+(i)} U_f\|_F^2 \\
&+ \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2, \tag{5}
\end{aligned}$$

$I_{ij}$  : Indicator Function

$\mathcal{F}^+(i)$  : the set of friends of user  $u_i$

$\alpha, \lambda_1, \lambda_2$  : hyper - parameter

$\|\cdot\|_F$  : Frobenius - norm

위 Objective를 살펴보면, 총 3가지 Term에 걸쳐 작성되어 있는 것을 확인할 수 있는데

### 1. 첫 번째 Term

- $user_i$ 가  $item_j$ 에 몇 점을 줬는지 아는  $R_{ij}$ 에 대해서 모델의 예측값과 차이를 측정하는 방식
- $I_{ij}$ 가 어떠한  $i, j$ 가 Rating 되었는지 유무를 알려줌

### 2. 두 번째 Term

- $user_i$ 의 Feature vector와 친구들의 평균적인 Feature vector와 가까워지도록 하는 방식
- 이 방식은 모든 유저들의 취향은 그 유저의 친구들의 평균에 가까울 것이라는 가정이 내포
- In real world에서 친구들의 특성이 다양할 수 있기 때문에 새로운 방식 제안
- $user_i$ 와  $friend_f$ 의 취향이 얼마나 가까운지를 기반으로 Similarity 계산하여 반영

$$\frac{\alpha}{2} \sum_{i=1}^m \left\| U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f)} \right\|_F^2,$$

### 3. 세 번째 Term

- Sparsity가 큰 Rating-matrix는 Overfitting에 민감하기 때문에 Regularization term 추가

위에서 언급한 것과 같이 비슷한 친구들은 더 가깝게, 먼 친구들은 적게 반영하는 방식을 통해 Model 1: Average-based Regularization Model 이 정의된다.

$$\begin{aligned} \min_{U, V} \mathcal{L}_1(R, U, V) &= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (R_{ij} - U_i^T V_j)^2 \\ &+ \frac{\alpha}{2} \sum_{i=1}^m \left\| U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} Sim(i, f)} \right\|_F^2, \\ &+ \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2. \end{aligned} \quad (8)$$

## Gradient Descent

$$\begin{aligned}
\frac{\partial \mathcal{L}_1}{\partial U_i} &= \sum_{j=1}^n I_{ij} (U_i^T V_j - R_{ij}) V_j + \lambda_1 U_i \\
&\quad + \alpha \left( U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} \text{Sim}(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} \text{Sim}(i, f)} \right) \\
&\quad + \alpha \sum_{g \in \mathcal{F}^-(i)} \frac{-\text{Sim}(i, g) \left( U_g - \frac{\sum_{f \in \mathcal{F}^+(g)} \text{Sim}(g, f) \times U_f}{\sum_{f \in \mathcal{F}^+(g)} \text{Sim}(g, f)} \right)}{\sum_{f \in \mathcal{F}^+(g)} \text{Sim}(g, f)}, \\
\frac{\partial \mathcal{L}_1}{\partial V_j} &= \sum_{i=1}^m I_{ij} (U_i^T V_j - R_{ij}) U_i + \lambda_2 V_j.
\end{aligned} \tag{9}$$

- 위와 같이 정의되는 Gradient를 이용해 SGD로 학습한다.

## Model 2: Individual-based Regularization

저자는 만약 User의 친구들이 다양한 취향을 가지고 있다면 이러한 특성이 잘 반영이 되지 않을 수도 있다는 가능성을 언급하며 새로운 Regularization 방법을 제안한다.

- Model 1 같은 경우 Similarity를 이용하긴 했어도  $user_i$ 와  $friend_j$ 가 결국 하나의 vector로 표현되어 짬뽕되기 때문에 제대로 반영하지 못한다는 문제점이 있다.
- 또한, Netflix 논문에서 언급된 문제점인 실제 유사한 유저가 존재하지 않아도 가중 평균 방식을 취하기 때문에 생각보다 더 반영될 수 있다는 문제점이 존재한다.
- 이러한 문제점을 해결하기 위해 식을 넓게 펼치는 느낌으로 아래와 같은 새로운 Term이 제안된다.

$$\frac{\beta}{2} \sum_{i=1}^m \sum_{f \in \mathcal{F}^+(i)} \text{Sim}(i, f) \|U_i - U_f\|_F^2,$$

## Gradient Descent

$$\begin{aligned}
\frac{\partial \mathcal{L}_2}{\partial U_i} &= \sum_{j=1}^n I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_1 U_i \\
&+ \beta \sum_{f \in \mathcal{F}^+(i)} Sim(i, f)(U_i - U_f) \\
&+ \beta \sum_{g \in \mathcal{F}^-(i)} Sim(i, g)(U_i - U_g), \\
\frac{\partial \mathcal{L}_2}{\partial V_j} &= \sum_{i=1}^m I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_2 V_j.
\end{aligned}$$

- 위와 같이 정의되는 Gradient를 이용해 SGD로 학습된다.

## Similarity Function

### 1. Vector Space Similarity (VSS)

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{fj}^2}},$$

- $user_i$ 와  $friend_f$ 가 서로 평가한  $item_j$ 의 Rating을 이용해 코사인 유사도 도출
- 이러한 방식은 User의 bias를 표현하지 못하는 문제점 존재한다.

### 2. Pearson Correlation Coefficient (PCC)

$$Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \bar{R}_i) \cdot (R_{fj} - \bar{R}_f)}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \bar{R}_i)^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{fj} - \bar{R}_f)^2}}, \quad (14)$$

- Pearson Correlation Coefficient를 통해 Similarity 계산
- PCC는 -1에서 1까지의 값을 갖지만 VSS는 0에서 1까지의 값을 가지기 때문에  $f(x) = (x + 1)/2$ 의 mapping function을 이용해 PCC의 Range를 bound 시킨다

## Extensions

분명 Abstract에서 "제안된 모델이 other contextual information을 어떻게 반영할 수 있는지"를 보여준다고 했는데 이 Section에서는 가능성만 언급하지 어떻게 결합하는지는 구체적인 논의는 없기 때문에 조금 의아한 점 존재한다

## Experimental Analysis

### Dataset

#### 1. Douban

- Chinese Web 2.0 providing user rating, review and recommendation services for movies, books and music

#### 2. Epinions

- customer review platform
- trust / distrust relationship

	Douban	Epinions
# of users	129,490	51,670
# of items	58,541	83,509
# of ratings	16,830,839	631,064
# of links	1,692,952	511,799

# Comparison

**Table 5: Performance Comparisons (Dimensionality = 10)**

Dataset	Training	Metrics	UserMean	ItemMean	NMF	PMF	RSTE	SR1 <sub>vss</sub>	SR1 <sub>pcc</sub>	SR2 <sub>vss</sub>	SR2 <sub>pcc</sub>
Douban	80%	MAE	0.6809	0.6288	0.5732	0.5693	0.5643	0.5579	0.5576	0.5548	<b>0.5543</b>
		Improve	18.59%	11.85%	3.30%	2.63%	1.77%				
		RMSE	0.8480	0.7898	0.7225	0.7200	0.7144	0.7026	0.7022	0.6992	<b>0.6988</b>
	60%	Improve	17.59%	11.52%	3.28%	2.94%	2.18%				
		MAE	0.6823	0.6300	0.5768	0.5737	0.5698	0.5627	0.5623	0.5597	<b>0.5593</b>
		Improve	18.02%	11.22%	3.03%	2.51%	1.84%				
	40%	RMSE	0.8505	0.7926	0.7351	0.7290	0.7207	0.7081	0.7078	0.7046	<b>0.7042</b>
		Improve	17.20%	11.15%	4.20%	3.40%	2.29%				
		MAE	0.6854	0.6317	0.5899	0.5868	0.5767	0.5706	0.5702	0.5690	<b>0.5685</b>
		Improve	17.06%	10.00%	3.63%	3.12%	1.42%				
		RMSE	0.8567	0.7971	0.7482	0.7411	0.7295	0.7172	0.7169	0.7129	<b>0.7125</b>
		Improve	16.83%	10.61%	4.77%	3.86%	2.33%				
Epinions	90%	MAE	0.9134	0.9768	0.8712	0.8651	0.8367	0.8290	0.8287	0.8258	<b>0.8256</b>
		Improve	9.61%	15.48%	5.23%	4.57%	1.33%				
		RMSE	1.1688	1.2375	1.1621	1.1544	1.1094	1.0792	1.0790	1.0744	<b>1.0739</b>
	80%	Improve	8.12%	13.22%	7.59%	6.97%	3.20%				
		MAE	0.9285	0.9913	0.8951	0.8886	0.8537	0.8493	0.8491	0.8447	<b>0.8443</b>
		Improve	9.07%	14.83%	5.68%	4.99%	1.10%				
		RMSE	1.1817	1.2584	1.1832	1.1760	1.1256	1.1016	1.1013	1.0958	<b>1.0954</b>
		Improve	7.30%	12.95%	7.42%	6.85%	2.68%				

- 비교 모델에서 social information을 올바른 방법으로 결합한 모델이 없기 때문에 SOTA trust-aware recommendation과 비교한다고 언급
- 모든 데이터셋에서 MAE, RMSE를 이용한 지표에서 비교 모델 중에서 가장 좋은 성능을 보이고 있음
- Model 1보다 Model 2, VSS보다 PCC를 이용한 모델이 성능이 더 좋음

## Impact of Parameters $\alpha$ and $\beta$

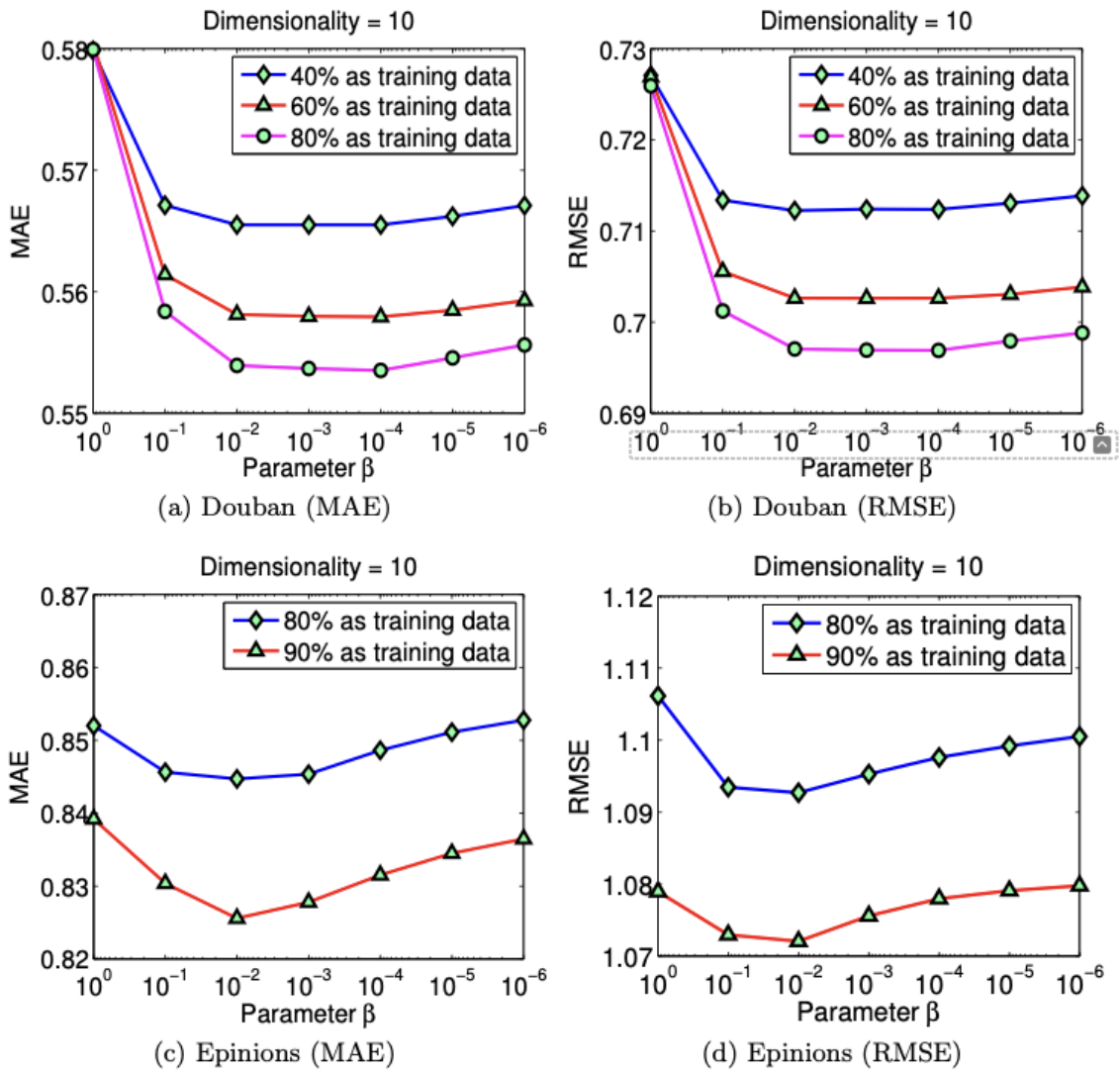


Figure 2: Impact of Parameter  $\beta$  (Dimensionality = 10)

- $\alpha$  and  $\beta$  모두 Social Network Information을 공유하는 정도를 정하는 하이퍼파라미터기 때문에 Metric 변화 양상이 비슷

## Impact of Similarity Functions



**Table 6: Similarity Analysis (Dimensionality = 10)**

Dataset	Training	Metrics	SR2 Sim=1	SR2 Sim=Ran	SR2 <sub>vss</sub>	SR2 <sub>pcc</sub>
Douban	80%	MAE	0.5579	0.5592	0.5548	0.5543
		RMSE	0.7034	0.7047	0.6992	0.6988
	60%	MAE	0.5631	0.5643	0.5597	0.5593
		RMSE	0.7083	0.7098	0.7046	0.7042
	40%	MAE	0.5724	0.5737	0.5690	0.5685
		RMSE	0.7195	0.7209	0.7129	0.7125
Epinions	90%	MAE	0.8324	0.8345	0.8258	0.8256
		RMSE	1.0794	1.0809	1.0744	1.0739
	80%	MAE	0.8511	0.8530	0.8447	0.8443
		RMSE	1.1002	1.1018	1.0958	1.0954

- Similarity가 전체적인 모델에 어떠한 영향을 끼치는지 분석하기 위해 Random-initialization 방식, 그리고 모든 Similarity를 1로 두는 방식을 추가하여 Metric을 비교하였음
- 실질적으로 Similarity가 모델에 영향을 끼치는 것으로 확인됨

## Conclusion and Future Work

### Conclusion

- ★★★★★: Social friend information을 incorporate 했음 !

### Limitation

#### 1. Suitable group problem

- real world에서 우리는 모든 사람들에게 물어보는 것이 아니라 그 분야에 expert에 대해 물어봄
  - ex) 영화에 대한 task일 경우 영화 매니아 또는 영화 전공자한테 물어봄
- 즉, most suitable group을 파악하는 것이 중요

#### 2. Item Interaction

- Item들간의 관계는 표현하지 못했음