# Introduction

**추천중요/ 약점(공격) / 방어2가지 / 방어의 약점과 그 이유 / 이유**

추천 시스템은 decision making에 큰 영향을 끼치는 것으로 생활 깊숙히 적용되는 중요한 분야이다. 추천 시스템은 사용자들의 행동을 기반으로 이루어지며 널리 사용되는 추천 시스템의 원칙은 비슷한 사람이 좋아했던 상품을 제시하는 것이다. 이런 원칙에 입각한 여러 방법들이 제시되었고 큰 성공을 거두었음.

하지만 이런 방법의 약점을 파악한 공격자들이 다양한 시도로 추천 시스템의 질을 떨어뜨리는 상황이 발생함.

CF는 그 본질상 여러가지 공격에 취약함 (random attack, bandwagon attack…)

이를 해결하기 위해 대표적으로 두가지 방법이 있음

첫째, abnormal user를 detect하여 해당 사용자를 제외하고 추천 알고리즘을 적용

둘째, 사용자가 남긴 review quality를 고려한 추천 알고리즘 적용

첫째 방법을 좀 더 자세히 말하면 review의 text data, meta data나 user-item으로 구성되는 graph-based feature로 outlier를 찾아내는 것으로 널리 연구되는 분야임. 이런 방법으로 아웃라이어를 검출하여 제외한뒤 추천 알고리즘을 적용하면 좋아지긴 함.

하지만 아웃라이어를 잡아내는것은 not perfect하며 대개 . 아웃라이어 검출 방법을 적용한 뒤라도 아웃라이어가 존재할 가능성이 있고 이는 추천 알고리즘을 degrade함

두번째 방법을 말하면 review quality를 고려하여 추천 알고리즘이 quality가 높은 review의 데이터에 높은 가중치를 두어 quality biased 추천을 하는 것임.

이 방법에서 가장 중요한 것은 review quality를 measure하는 방법임.

여러 사이트들은 사용자들에게 상품에 대한 리뷰를 남길 수 있게 하는 동시에 다른 사용자가 남긴 상품 리뷰에 관해 도움이 되는 리뷰인지에 대한 평가를 남길 수 있도록 함. 즉, 스마트한 공격자는 자신의 리뷰 퀄리티를 높이기 위해 shiril인가 뭐시기 공격을 할 수 있으므로 review quality를 measure는 robust해야함

예로 toy example을 들겠삼 100명 이상의 fake user가 target item 10개의 추천 평판을 높이기 위해 1000개의 fake review과 각 review에 도움이 되었다는 fake vote를 100개 하는 경우를 생각해보셈

견고한 퀄리티 측정을 하기 위해서 우린 다음과 같은 가정을 한다.

리뷰 퀄리티를 투표하는 사용자의 그룹이 다양하면 해당 리뷰가 객관적이라고 할 수 있을 것이다. 즉, 투표자 수가 아닌 투표자 그룹의 수가 많을 수록 객관적인 리뷰라고 보는 것이다. (미국 선거 예로 들까?)

거꾸로 생각하면 동일한 취향을 가진 user 100명이 도움이 되었다고 평가한 리뷰 a과 각기 다른 취향을 가진 user 50명이 도움이 되었다고 평가를 한 리뷰 b가 있을 때, 특정 그룹만을 위한 리뷰보단 전반적인 그룹에 도움이 되는 리뷰가 더 도움이 된다고 말할 수 있다.

방어 관점에서 보면 fake user는 동일한 그룹에 소속될 것이므로 fake user가 행한 vote의 영향력을 낮출수 있게 된다.

이러한 점을 고려하여 이 논문에선 review quality를 robust하게 측정하고 quality biased 된 추천 알고리즘을 제안함. 구체적으로 말하면 3가지 스텝이 있는데 첫번째는 사용자의 특징 벡터를 추출한다. 특징 벡터를 추출하기 위해선 사용자의 review history와 review rating history를 사용한다. 두번째 스텝에선 특정 사용자가 특정 아이템에 대해 남긴 리뷰의 퀄리티를 측정한다.

# Related Work

Try to find outlier using Rating matrix

Try to find outlier using Rating matrix, review text, and so on.

Robust Collaborative Filtering…

Recently, some researchers focus on review helpfulness.

Review-quality aware recommendation [] proposed estimation of helpfulness value of review and new objective function including review’s helpfulness score.

DualRec[] exploit helpfulness vote data to reliberate cold start and data sparsity problem.

These studies assume helpfulness vote data originate from genuine users. In most recommender system, their assumptions are not practical. Attacker can inject not only fake reviews but also fake helpfulness votes.

# Motivation

Imagine that

# Proposed Method

Step 1) Embed users on latent space

Step 1-1) (reviewer-reviewer) , (reviewer-rater), (rater-rater) enumeration

Step 1-2) word2vec 알고리즘의 input으로 넣음

Step 2) Measure helpfulness of reviews

Step 2-1) helpfulness(review(reviewer,item)) = sum(c\*exp(-r\*emb(reviewer)\*emb(rater)))/|rater set|

//공감은 reviewer랑 rater가 다른 그룹이면 좋은데/ 비공감은 reviewer랑 rater가 같아야 좋은거아닐까????

//Step 2-2) helpfulness credibility : rater 수?

Step 3) Apply collaborative filtering

Cost = ||H\*(R-UV)||\_f

# Experiment

이 장에선 제안한 방법의 견고함을 보이기 위한 실험과 그 결과를 서술한다.

리뷰 데이터 셋에 공격을 가하기 전과 후의 결과를 비교함으로써 추천 알고리즘의 견고함을 보인다.

공격 시나리오 setup은 다음과 같다.

Genuine review and vote

Bandwagon Attack // Random Attack // Average Attack

To choose items to attack, we use the following filter: an item which has not been voted by more than 5% of the user population and has an average vote of less than 3 (since our data set has votes between 1–5).

We then vary the number of profiles inserted and the number of items voted by the spam user (filler size).

Target\_item : badly rated item or newly injected item

공격자가 선택한 타겟 아이템의 평점이 얼마나 변화하는지에 따라 추천 알고리즘의 견고함을 측정한다.

1. 타겟 아이템의 관측된 데이터
2. 공격이 없는 데이터를 입력으로 받은 추천 알고리즘의 타겟 데이터에 대한 평점 예측 결과
3. 공격자의 데이터가 추가된 데이터를 입력으로 받은 추천 알고리즘의 타겟 데이터에 대한 평점 예측 결과

견고한 알고리즘은 공격이 있는 상황에서도 보통 사용자의 타겟 아이템에 대한 평점을 well fit하고 공격자의 타겟 아이템에 대한 평점을 fit하지 않아야 한다.

제안한 방법과 비교할 추천 알고리즘은 2가지이다. Base MF, naïve review quality aware MF이다.

Base MF는 모든 review의 weight를 동일하게 간주하여 MF를 하는 방법으로 식 #에서 W matrix가 전부 1인 것으로 볼 수 있다. Naïve review quality aware MF는 review의 helpfulness vote 데이터를 신뢰하여 계산한 weight로 W를 구성하여 식 #을 최적화하는 방법이다.

Metric (robustness)

Shift of target item: MAE=|prediction(target\_items) - observed(target\_items)|

**MAE on the rating target items by the normal users**: to measure the effect of the attack, we compute MAE

**Prediction Shift** measures the change in prediction of the attacked item (before and after attack) of a CF algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| **절대값도 구하고**  **차이도 구하고** | **Base MF** | **Naïve RQMF** | **My MF** |
| **공격 전 RMSE** | 0.5 | 0.4 | **0.3** |
| **공격 후 RMSE** | 1.5 | 1.3 | **0.5** |
| **Prediction shift**  **(공격 전-공격 후 차이)** | 1.0 | 0.9 | **0.2** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attacker size** | **Target size** | **Filler size** | **Base MF** | **Naïve RQMF** | **My MF** |
| **1** | **1** | **1** | **전/후/차이** |  |  |
| **1** | **1** | **5** |  |  |  |
| **1** | **1** | **10** |  |  |  |
| **5** | **1** | **1** |  |  |  |
| **5** | **1** | **5** |  |  |  |
| **5** | **1** | **10** |  |  |  |
| **5** | **5** | **5** |  |  |  |
| **5** | **5** | **10** |  |  |  |
|  |  |  |  |  |  |

**Comparison of Helpfulness between fake and normal reviews ( 1 1 10 / 5 1 10)**

|  |  |  |  |
| --- | --- | --- | --- |
| Helpfulness | **Base MF** | **Naïve RQMF** | **My MF** |
| 90 percentile (top 10%) | 1 | 5 | 4 |
| 50 percentile (top 50%) | 1 | 4 | 3 |
| 10 percentile (top 90%) | 1 | 2 | 2 |
| Mean(helpfulness(fake)) | 1 | **5** | **2.5** |
| Fake/top 10 ratio |  |  |  |
| Fake/top 50 ratio |  |  |  |
| Fake/top 90 ratio |  |  |  |

Metric (accuracy)

**Overall prediction: RMSE on the rating dataset without the profiles of attacker**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Base MF** | **Naïve RQMF** | **My MF** |
| **공격 전 RMSE** | 1.00 | 0.95 | **0.90** |
| **공격 후 RMSE** | 1.5 | 1.3 | **0.95** |
| **차이** | 0.5 | 0.35 | **0.05** |

Result

Shift of target item: MAE=|prediction(target\_items) - value(target\_items)|

Not yet

Overall prediction: RMSE without attacker

Not yet

Comparison of Helpfulness between fake and normal reviews

('fake helpful mean', 2.5045178962777706)

('25', 3.0000000000000013, 0.83483929875925644)

('50', 3.9999997046891473, 0.62612952029515323)

('75', 4.0, 0.62612947406944264)

('90', 4.0, 0.62612947406944264)