Adversarial Attacks
on an oblivious recommender
Why care for adversarial attacks in recommender systems?
How vulnerable are recommendation models to machine learned adversarial attacks?
Form of Adversarial Attacks in recommendation systems
Form of Adversarial Attacks in recommendation systems
Two Lines of Prior Work

1

“Shilling Attacks” in Recommender Systems

Hand-Engineered fake user profiles in Recommendation Systems
Two Lines of Prior Work

1. “Shilling Attacks” in Recommender Systems
   Hand-Engineered fake user profiles in Recommendation Systems

2. Adversarial Examples
   Learned Adversarial Attacks in other domains

   ![Images](School bus, Perturbation, Ostrich)
Combine both approaches:
revisit adversarial attacks on recommenders
from a machine learned optimization perspective
Challenges specific to the recommendation setting

Collaborative Filtering cascading effects?
Challenges specific to the recommendation setting

Collaborative Filtering cascading effects?

Un-noticeability of attacks
Challenges specific to the recommendation setting

a) Collaborative Filtering cascading effects?
b) Un-noticeability of attacks
c) Need to learn model iteratively
Poisoning attacks
Challenges specific to the recommendation setting

- No access to gradient
- Need to learn model iteratively
- Collaborative Filtering cascading effects?
- Un-noticeability of attacks
- Poisoning attacks
Formulating the problem

Recommender System

Adversary

Two-player general-sum min-max game:

\[
\theta_{R}^{t+1} = \operatorname{argmin}_{\theta_R} f_R(\theta_R, Z^t)
\]

\[
Z^{t+1} = \operatorname{argmin}_{Z} f_A(\theta_{R}^{t}, Z)
\]
Assumptions

Recommender System

Oblivious to the existence of the adversary
Assumptions

Recommender System

Can evaluate how incorporating the fake users would change the recommender’s scores

- Knows R’s loss function
- Knows R’s parametric representation
- Cannot evaluate R’s gradient
Recommender System

\[
\min_{\theta_R} f_R(\theta_R, Z) = \min_{\theta_R} \frac{1}{C_{all}} \sum_{c=1}^{C_{all}} \ell(y_c, \hat{y}_c(u_c, i_c; \theta_R, Z))
\]

Fit my model on data.
Recommender System

Adversary

Create fake user matrix Z.
Recommender System

Fit my model on data.

Adversary
Recommender System

Adversary

Create fake user matrix $Z$ until I achieve my goal

$Z_{t+1}$
Adversary’s Goals

**Goal 1:** Create fake users so that they are *indistinguishable* from real users

**Goal 2:** Create fake users so that they achieve an *adversarial* intent
Adversary’s Goals

Goal 1: Create fake users so that they are indistinguishable from real users

The idea

Distribution-preserving adversarial users

Minimize Jensen-Shannon divergence among real-fake distributions

Generative Adversarial Networks are a great fit.

First stage of attacker strategy
Adversary’s Goals

Goal 2: Create fake users so that they achieve an adversarial intent

Projected gradient descent:

\[
\tilde{Z}_{t+1} = Z_t - \eta \nabla_{Z_t} f_A(Z) \\
Z_{t+1} = \Pi_{\text{allowed range}}(\tilde{Z}_{t+1})
\]
How to obtain the gradient?

**Challenges:**
- Learn recommender iteratively after injecting fake user profiles
- **Bandit** feedback, no access to gradient

---

**Adversary’s Goals**

**Goal 2:** Create fake users so that they achieve an **adversarial intent**
**Adversary’s Goals**

**Goal 2:** Create fake users so that they achieve an adversarial intent

---

**How to obtain the gradient?**

**Idea:** Query Recommender on K directions to construct gradient approximation

\[
\nabla f_A(Z_t) = \frac{1}{\alpha} \sum_{h=1}^{K} (f_A(Z_t + \alpha Z^{(h)}) - f_A(Z_t))Z^{(h)}
\]
The adversary removes the target item from the target user’s top-10 list.
Targeting Item’s Mean Predicted Score

$$\Delta(Z) = f_{\text{before}}(X) - f_A(X; Z)$$

Each user’s $\Delta(Z)$

This is a **hard task** for the adversary.
Targeting the top User of an Item

\[ \Delta(Z) = f_{\text{before}}(X) - f_{\text{A}}(X; Z) \]

Targeting the top user also targets all top-K users for the target item.
Takeaways

Proposed **general approach** for ML adversarial attacks to recommender systems

Considered **new types of attacks**

Novel algorithm using 0th order optimization, as **no access** to the gradient & iterative procedure

Effective attacks show the **need for adversary-aware recommenders**

Thank you!