PORE: Provably Robust Recommender Systems against Data Poisoning Attacks

Jinyuan Jia¹, Yupei Liu², Yuepeng Hu², Neil Zhenqiang Gong²
¹Penn State University
²Duke University
08/09/2023
The first two authors made equal contribution.
Recommender Systems

• Widely deployed to engage users
  • Amazon, YouTube, TikTok, eBay

• Recommender system
  • Input: Rating-score matrix
  • Output: Recommended top-$N$ items for each user

• Recommender system algorithm
  • Bayesian Personalized Ranking (BPR)
  • Item-based Recommendation (IR)
  • Neural Collaborative Filtering (NCF)
Recommender Systems

<table>
<thead>
<tr>
<th></th>
<th>$i_1$</th>
<th>$i_2$</th>
<th>$i_3$</th>
<th>$i_4$</th>
<th>$i_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$u_2$</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$u_3$</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$u_4$</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Rating-score matrix

<table>
<thead>
<tr>
<th></th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
<th>$u_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>${i_5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td>${i_5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td>${i_1}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_4$</td>
<td>${i_3}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recommender system algorithm

Top-1 recommendation
Recommender Systems are Vulnerable to Data Poisoning Attacks

• An attacker could inject fake users
  • By registering and maintaining fake accounts

• At most e fake users
  • Give an arbitrary rating score to an item
  • Rate as many items as the fake user wishes

• A poisoned recommender system makes attacker-desired, arbitrary recommendations.
Recommender Systems are Vulnerable to Data Poisoning Attacks

\[
\begin{align*}
\mathbf{u}_1 &= \begin{pmatrix} 5 & 4 & 5 & 0 & 0 \end{pmatrix} \\
\mathbf{u}_2 &= \begin{pmatrix} 4 & 0 & 5 & 0 & 0 \end{pmatrix} \\
\mathbf{u}_3 &= \begin{pmatrix} 0 & 0 & 5 & 1 & 4 \end{pmatrix} \\
\mathbf{u}_4 &= \begin{pmatrix} 4 & 2 & 0 & 0 & 4 \end{pmatrix} \\
\mathbf{u}_5 &= \begin{pmatrix} 0 & 4 & 5 & 4 & 0 \end{pmatrix}
\end{align*}
\]

Poisoned rating-score matrix

Recommender system algorithm

\[
\begin{align*}
\mathbf{u}_1: \{i_4\} \\
\mathbf{u}_2: \{i_5\} \\
\mathbf{u}_3: \{i_2\} \\
\mathbf{u}_4: \{i_3\} \\
\mathbf{u}_5: \{i_1\}
\end{align*}
\]

Top-1 recommendation
Limitations of Existing Defenses

- **Empirical defenses**
  - Cannot provide formal robustness guarantee

- **Provable defenses**
  - Designed for classifiers: Bagging
  - Suboptimal provable robustness guarantees
PORE: First Framework to Build Provably Robust Recommender Systems

• Create multiple sub-rating-score matrices
  • Each sub-rating-score matrix: rating scores of $s$ randomly sampled users

• Build a base recommender system upon each sub-rating-score matrix
  • Use an arbitrary recommender system algorithm

• Build an ensemble recommender system
  • Majority vote
An Example for PORE

Poisoned-rating-score Matrix

Top-1 Recommendation

Majority Vote
The Provable Robustness Guarantee of PORE

\[ \mathcal{E}_u \]  A set of ground-truth items for a user \( u \)

\[ \mathcal{L}(M, e) \]  A set of all possible poisoned rating-score matrices

With a probability at least \( 1 - \alpha \), we have:

\[
\min_{M' \in \mathcal{L}(M, e)} |\mathcal{E}_u \cap \mathcal{A}(M', u)| \geq r_u
\]
Computing the Robustness Guarantee

• Formulating the computation of $r_u$ as the following optimization problem:

$$r_u = \arg \max_{r' \in \{1, 2, \ldots, \min(k, N)\}} r'$$

$$s.t. \quad p_{\mu_{r'}}^* > \min \left( \min_{c=1}^{N-r'+1} \frac{N' \cdot (\overline{p}_{H_c}^* + \sigma)}{c} \right), \overline{p}_{v_1}^* + \sigma$$
Recommender System Setup

• MovieLens-1M
  • 1,000,209 rating scores
  • 6,040 users and 3,952 items

• Base recommender system algorithm
  • BPR

• Parameter setting
  • $N'=1$ (number of items recommended by a base recommender system)
  • $N=10$ (number of items recommended by our ensemble recommender system)
  • $T=100,000$ (total number of base recommender systems)
  • $\alpha=0.001$ (1- $\alpha$ is the confidence score)
  • $s=500$ (number of users in each sub-rating-score matrix)
Evaluation Metrics

• Precision@$N$
  • The fraction of recommended items that are in the ground truth set of a user

• Recall@$N$
  • The fraction of items in the ground truth set that are recommended

• F1-Score@$N$
  • Tradeoff between Precision@$N$ and Recall@$N$
Evaluation Metrics

Certified Precision@N = \frac{r_u}{N}

Certified Recall@N = \frac{r_u}{|E_u|}

Certified F1-Score@N = \frac{2 \cdot r_u}{|E_u| + N}
PORE is Provably Robust against Data Poisoning Attacks
PORE Maintains Utility

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision@10</th>
<th>Recall@10</th>
<th>F1-Score@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPR</td>
<td>0.324449</td>
<td>0.118385</td>
<td>0.144765</td>
</tr>
<tr>
<td>Ensemble BPR</td>
<td>0.362945</td>
<td>0.119441</td>
<td>0.151509</td>
</tr>
</tbody>
</table>

Our PORE maintains utility without attacks.
Time Complexity of PORE
Compared Method

• Bagging
  • State-of-the-art method to build provably robust classifier
PORE Outperforms the Existing Method
Summary

• We propose the first framework to build provably secure recommender systems

• Our PORE could be applied to an arbitrary recommender system algorithm

• Our PORE outperforms existing method extended from classifiers