Privacy-preserving movie recommendation mechanism: Top N recommendations with PNCF on perturbed user data

Mingyang Sun

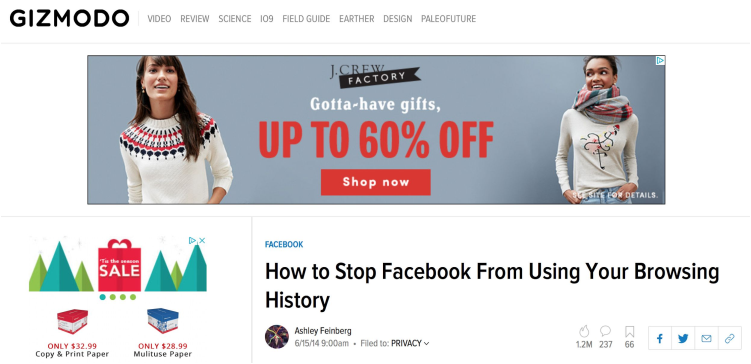
Acknowledgement: The prototype algorithm is from the paper:

Differential Privacy for Neighborhood-based Collaborative Filtering

Author(s): Tianqing Zhu, Gang Li, Yongli Ren, Wanlei Zhou, Ping Xiong

2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Motivation

- Recommender systems are applied worldwide in e-commerce
    - Netflix, YouTube, Taobao, Facebook Ads, Google Ads...
- Privacy issues remain controversial
    - Netflix canceled privacy sequel after lawsuit
    - 

Prototype Algorithms used by RS

- Collaborative Filtering (CF)
- Matrix Decomposition
- Clustering
- Deep Learning Approaches

And more...

Prototype Algorithms used by RS

- Collaborative Filtering (CF) **<- popular approach, but vulnerable to privacy attacks**
- Matrix Decomposition
- Clustering
- Deep Learning Approaches

And more...

Traditional Solutions with DP

· McSherry et al.

“Differentially private recommender systems: Building privacy into the netflix prize contenders”

Private covariance matrix to randomize each user’s rating before submitting to the system

· Machanavajjhala et al.

“Personalized social recommendations: accurate or private”

Graph link-based recommendation algorithm

Pitfall 1: Fail to Hide Similar Neighbors - KNN attack

- k Sybil users
- Inference from recommendations

User 1	1	5	3	4	3
User 2	4	1	5	2	5
User 3	2	5	3	5	4

Pitfall 2: Too Large Noise for Recommendation

· High sensitivity

Queries employed have high sensitivities.

· Naive mechanism

Previous work disregards characteristics of recommendation systems.

Private Neighbor Collaborative Filtering (PNCF)

Two private operations

- **Private Neighbor Selection**

  exponential mechanism

  find k neighbors Nk(ti) on item similarity matrix

- **Perturbation**

  mask the ratings

  employ zero mean Laplace noise

Algorithm

Algorithm 1 Private Neighbor Collaborative Filtering(*PNCF*)

Input: **R**, privacy parameter ϵ, truncated parameter w, number of neighbors k, u_a, t_i, I

Output: $\hat{r}_{ai}$

1. Compute item to item similarity Matrix S;
2. Private Neighbor Selection: select k neighbors $N_k(t_i)$ from I;

rating

3. Perturbation: Perturb the similarity in $N_k(t_i)$ by adding $Lap(\frac{2k \cdot LS(i,\cdot)}{\epsilon})$ noise;
4. Predict $\hat{r}_{ai}$;

Private Neighbor Selection

- Naive exponential mechanism is too general
- Solutions:
    - Recommendation-aware sensitivity with smooth bound B
    - Truncated similarity in private neighbor selection

Recommendation-aware sensitivity with bound

- Recommendation-aware sensitivity (adapted based on LS)

$$RS(i,j) = \max_{i,j \in I} ||s(i,j) - s'(i,j)||_1.$$

- Apply smooth bound to the sensitivity to reduce noise

$$B(RS(i,j)) = exp(-\beta) \cdot RS(i,j).$$

Truncated Similarity in Private Neighbor Selection

- Score function used to enhance the quality of selected neighbors

$$\hat{s}(i,j) = \max\left(s(i,j), s_k(i,\cdot) - w\right)$$

- C1: all items whose similarities are larger than
- C0: the rest of items

$$\exp\left(\frac{\epsilon \cdot s(i,j)}{4k \cdot RS(i,j)}\right)$$

Private Neighbor Selection Algorithm

Input: ϵ, k, w, t_i, I ,$\mathbf{s}(i)$
Output: $N_k(t_i)$

```
Sort the vector s(i);
C_1 = [t_j | s(i,j) ≥ s_k(i,j) − w, t_j ∈ I],
C_0 = [t_j | s(i,j) < s_k(i,j) − w, t_j ∈ I],
for N=1:k do
    for each item t_j in t_i do
        Allocate probability as:
```

$$\frac{\exp\left(\frac{\epsilon\cdot\hat{s}(i,j)}{4k\cdot RS(i,j)}\right)}{\sum_{j\in C_1}\exp\left(\frac{\epsilon\cdot\hat{s}(i,j)}{4k\cdot RS(i,j)}\right)+|C_0|\cdot\exp\left(\frac{\epsilon\cdot\hat{s}(i,j)}{4k\cdot RS(i,j)}\right)}.$$

```
    end for
    Sample an element t from C_1 and C_0 without replacement according to probability;
    N_k(t_i) = N_k(t_i) + t;
end for
```

Utility Analysis

Suppose we randomly choose a small constant ρ less than 1, the following theorems could be proven.

- Theorem 1:

all $\rho > 0$, with probability at least $1 - \rho$, the similarity of all the items in $N_k(t_i)$ are larger than $s_k - w$, where $w = \min(s_k, \frac{4k \cdot RS}{\epsilon} \ln \frac{k \cdot (|v| - k)}{\rho})$.

- Theorem 2:

Theorem 3.2: Given an item t_i, for all $\rho > 0$, with probability at least $1 - \rho$, the similarities of all neighbors $> s_k + w$ are present in $N_k(t_i)$, where $w = \min(s_k, \frac{4k \cdot RS}{\epsilon} \ln \frac{k \cdot (|v| - k)}{\rho})$.

Utility Analysis

Takeaway:

No item in selected neighbors has similarity less than $(s_k(i,\cdot) + w)$ and every item whose similarity is greater than $(s_k(i,\cdot) + w)$ is selected.

Privacy Analysis

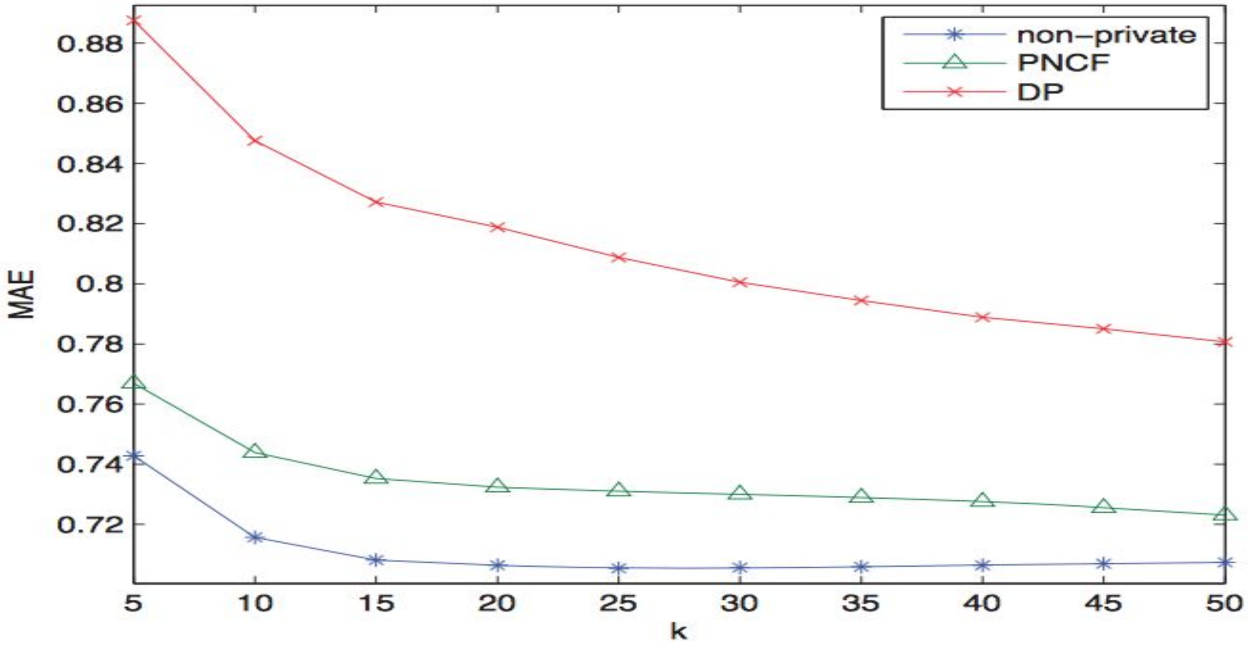
Each selection round: $\left(\frac{\epsilon}{2k}\right)$

Privacy Neighbor Selection: $\left(\frac{\epsilon}{2}\right)$.

Perturbation: Laplace noise added to the Nk(ti) set $\left(\frac{\epsilon}{2}\right)$

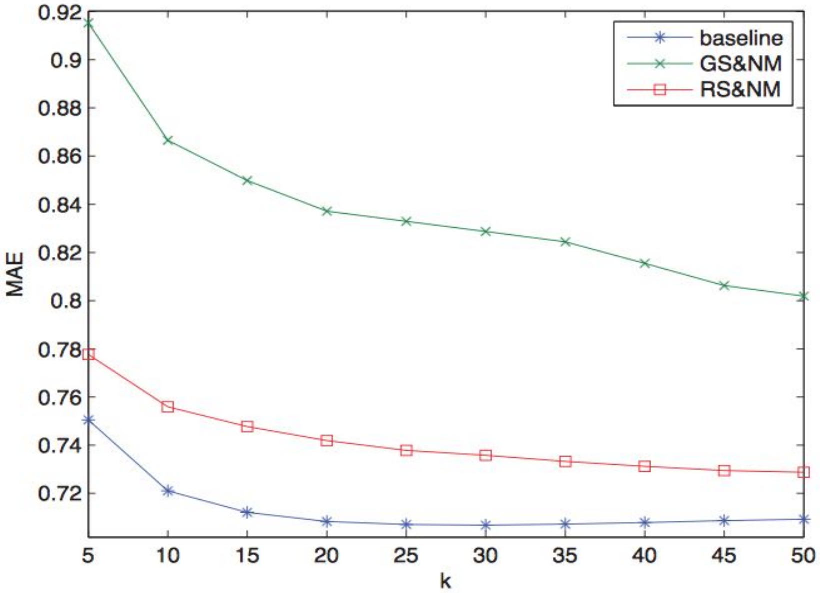
*Based on composition property of differential privacy

Performance - PNCF vs. DP

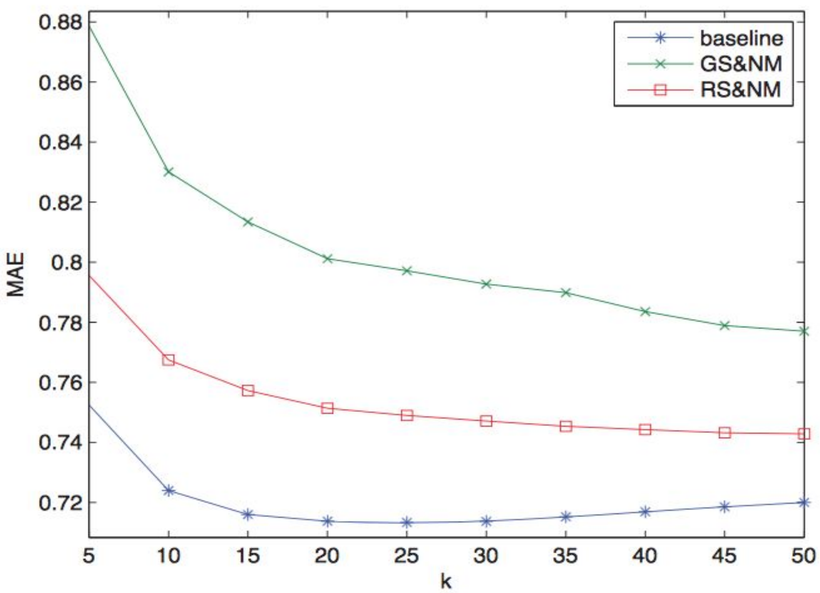


(a) PCC-item

Performance - GS vs. RS

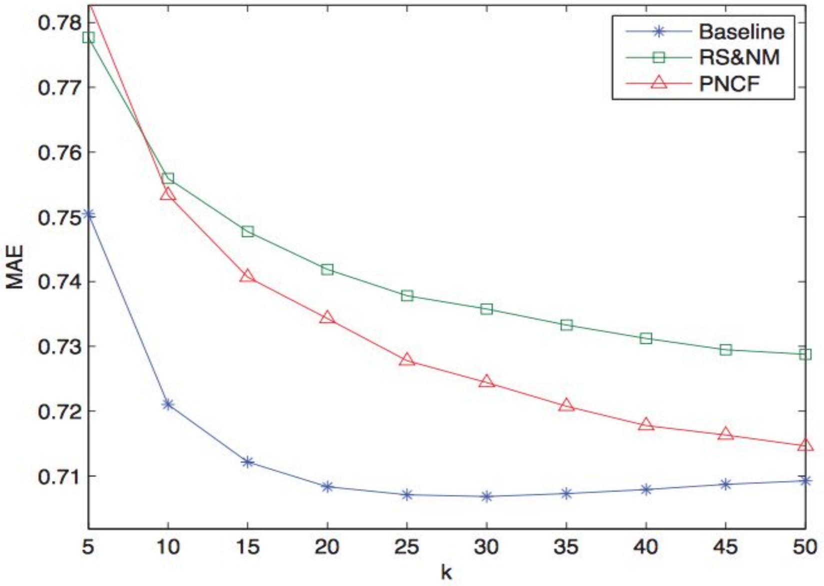


(a) PCC-item

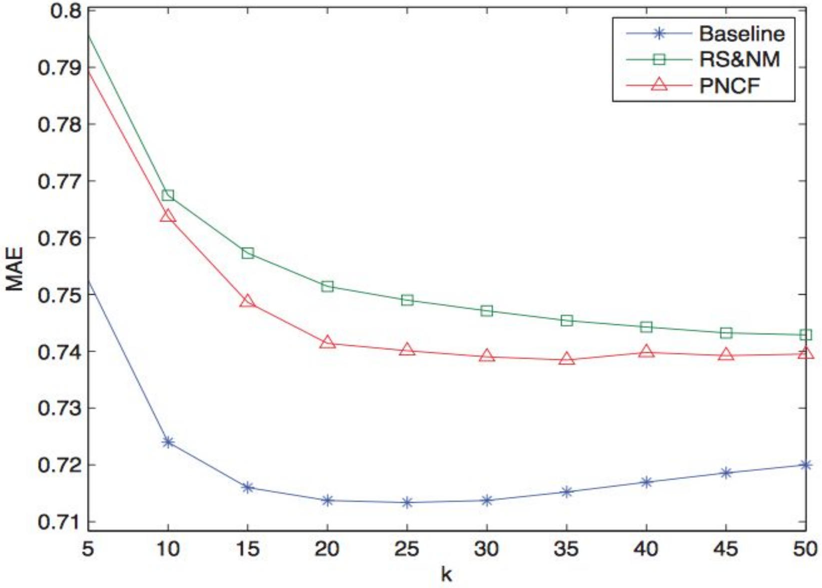


(b) COS-item

Performance - Naive vs. Improved EM



(a) PCC-item



(b) COS-item

Q&A

Thank you.