AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning

Jinyuan Jia, Neil Zhenqiang Gong

Department of Electrical and Computer Engineering

Iowa State University

OUTLINE

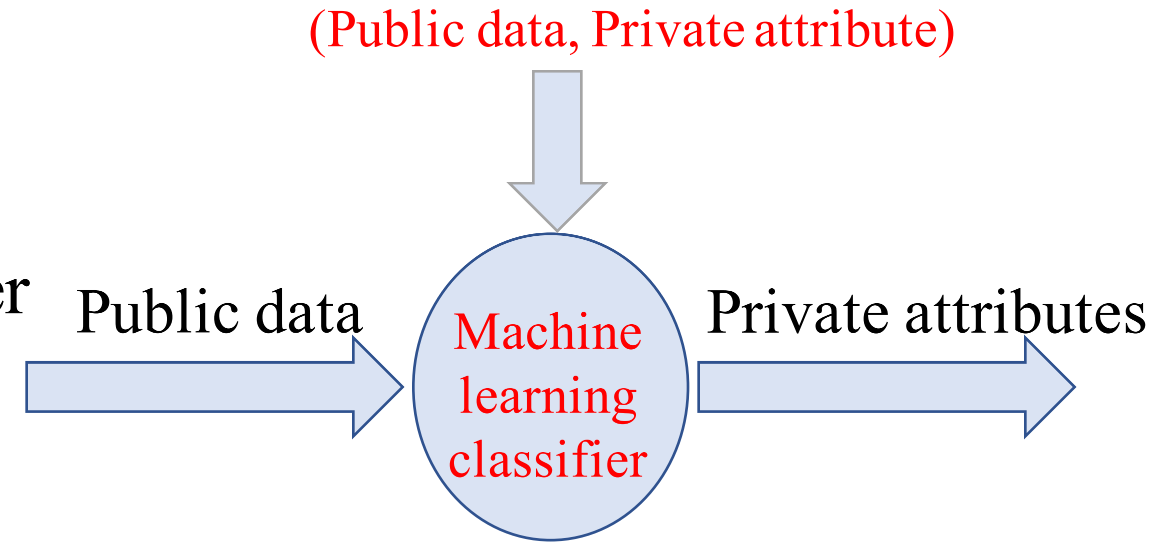
- Motivation
- Algorithm
- Evaluation
- Conclusion

OUTLINE

- Motivation
- Algorithm
- Evaluation
- Conclusion

Attribute Inference Attacks

- Input: User's public data
- Output: User's private attributes
- E.g. In social media, attacker can use machine learning classifier to infer user's private attributes.
  - Cambridge Analytica



- Private attributes and public data are statistically correlated

Attribute Inference Attacks are Pervasive

- Recommender systems
  - Public: Rating scores
  - Private: Gender
- Mobile apps
  - Public: User’s smartphone’s aggregate power consumption
  - Private: Locations
- Website fingerprinting
  - Public: Network traffic
  - Private: Websites
- Side-channel attacks
  - Public: Power consumption, processing time
  - Private: Cryptographic keys

Existing Defenses

- Game-theoretic methods:
  - Pros: Defend against optimal inference attacks
  - Cons: Computationally intractable
- Heuristic methods:
  - Pros: Computationally tractable
  - Cons:
    - Large utility loss
    - Direct access to user's private attribute value
- Local Differential Privacy (LDP)
  - Pros: Rigorous privacy guarantee
  - Cons: Large utility loss

Our Defense: AttriGuard

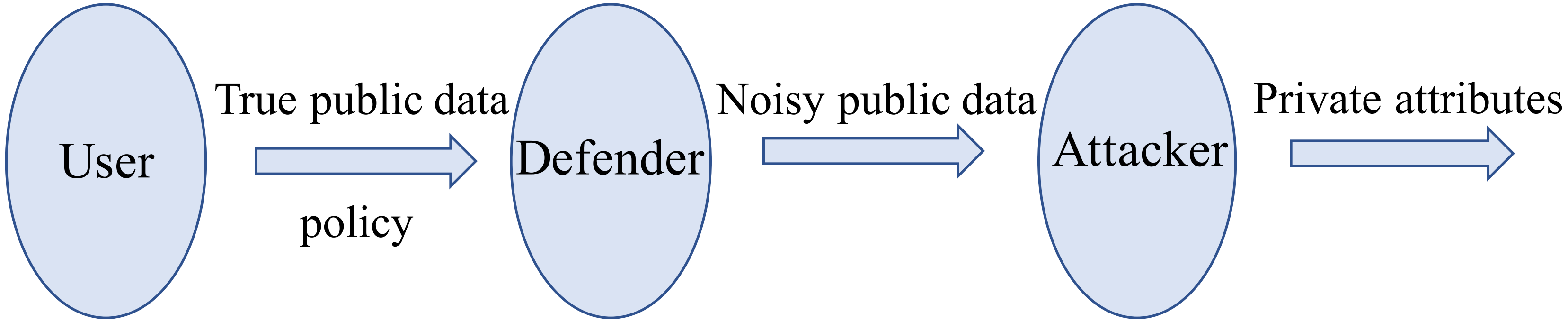
- Computationally tractable

- Small utility loss

OUTLINE

- Motivation
- Algorithm
- Evaluation
- Conclusion

Threat Model



- Policy A: Modify_Exist
- Policy B: Add_New
- Policy C: Modify_Add

Challenges

- The defender doesn't know the attacker's classifier C_a
  - The defender itself learn a classifier C
  - Transferability: similar classification boundaries
- Defender has no access to user's true private attribute value
  - Find a mechanism to add random noise
  - Output distribution of defender's classifier approaches certain *target probability distribution* that defender desires

Metric

- Difference between output distribution of defender's classifier $\mathbf{q}$ and *target probability distribution* $\mathbf{p}$
  - KL-divergence: $KL(\mathbf{p} \| \mathbf{q}) = \quad {}_i\, p_i \log \frac{p_i}{q_i}$
- Utility loss:
  - L_0 norm: $d(\mathbf{x}, \mathbf{x}+\mathbf{r}) = \|\mathbf{r}\|_0$

user's true public data vector

user's noisy public data vector

noise vector

Attribute-inference-attack Defense Problem

- Input:
  - *noise-type-policy*
  - *utility-loss-budget*
  - *target probability distribution*
  - *defender's classifier*
  - *user's true public data.*
- Output: *Mechanism* M that adds random noise
  - $M^*(\mathbf{r} \mid \mathbf{x})$ is the conditional probability that defender will add noise $\mathbf{r}$ to user's true public data $\mathbf{x}$
  - Sample from M to add noise

Attribute-inference-attack Defense Problem

$$M^{*} = \arg\min_{M} KL(\mathbf{p} \,||\, \mathbf{q})$$

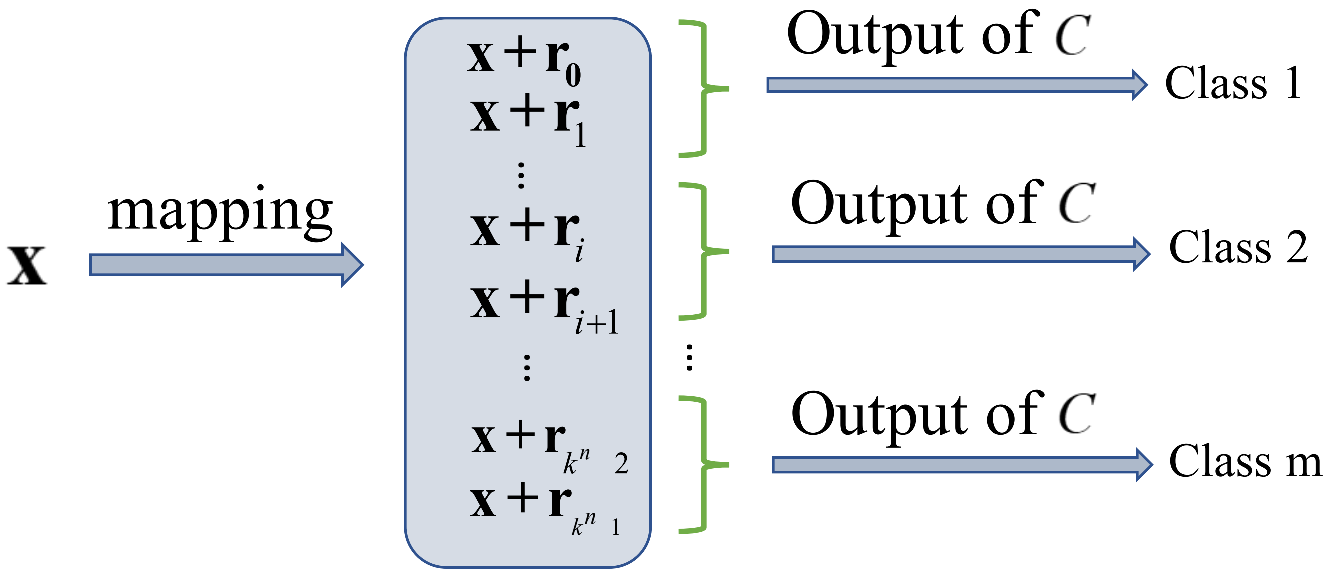
$$\textit{subject to} \quad E(d(\mathbf{x}, \mathbf{x}+\mathbf{r})) \le \beta$$

- $\mathbf{q}$:output distribution of defender's classifier C

$$q_i = \Pr(C(\mathbf{x}+\mathbf{r}) = i) = \underset{\mathbf{r}|C(\mathbf{x}+\mathbf{r})=i}{\qquad} M(\mathbf{r} \,|\, \mathbf{x})$$

Overview of AttriGuard

- Challenge to solve the optimization problem:
  - The probabilistic mapping $\mathbf{x} \rightarrow \mathbf{x}+\mathbf{r}$ is *exponential* to the dimensionality of $\mathbf{x}$
  - Categorize noise space into m groups to solve the challenge



Two-Phase Framework

- Phase I: For each noise group, find a minimum noise as representative noise

- Phase II: Simplify the mechanism M^* to be a probability distribution over m representative noise

Phase I

- Find minimum noise $\mathbf{r_i}$ for each group such that defender's classifier outputs class i given noisy public data input

$$\mathbf{r_i} = \arg\min_{\mathbf{r}} \|\mathbf{r}\|_0$$

$$\textit{subject to} \quad C(\mathbf{x} + \mathbf{r}) = i$$

Phase I

- The optimization problem can be viewed as *evasion attacks* to the defender's classifier

- Existing *evasion attacks* are insufficient
  - Not consider different *noise-type-policy*

- We propose PANDA based on *Jacobian-based Saliency Map Attack* (JSMA)
  - Consider *noise-type-policy*
  - Some entries in user's public data can be decreased while other entries can be increased in PANDA while all entries can either be increased or decreased in JSMA

Phase II

- Transform original optimization problem into following convex optimization problem:

$$M^* = \arg\min_{M} KL(\mathbf{p} \,||\, M)$$

subject to

$$\sum_{i=1}^{m} M_i \|\mathbf{r}_i\|_0 \le \beta$$

$$M_i > 0, \forall i \in \{1, 2, \ldots, m\}$$

$$\sum_{i=1}^{m} M_i = 1$$

M is a probability distribution, and M_i denote the probability select noise $\mathbf{r_i}$

OUTLINE

- Motivation
- Algorithm
- Evaluation
- Conclusion

Evaluation Dataset

- A review dataset from Gong and Liu (USENIX Security'16)
- Attributes considered: 25 cities
- Basic statistics

#Users	#apps	#ave. apps
16,238	10,000	23.2

- Training and Testing:
  - Training: 90% of users
  - Testing: the remaining users

Attribute Inference Attacks

- Defense unaware attack
  - Baseline attack (BA-A)
  - Logistic regression (LR-A)
  - Random forest (RF-A)
  - Neural network (NN-A)
- Robust classifier
  - Adversarial training (AT-A)
  - Defensive distillation (DD-A)
  - Region-based classification (RC-A)
- Detect noise
  - Detect noise via low-rank approximation (LRA-A)

Inference Accuracy without Defense

Attack	Inference Accuracy
BA-A	0.10
LR-A	0.43
RF-A	0.44
NN-A	0.39
AT-A	0.39
DD-A	0.40
RC-A	0.38
LRA-A	0.27

Defender's Classifier

- Neural Network (NN-D)
    - Use a different neural network architecture from attacker
- Logistic Regression (LR-D)

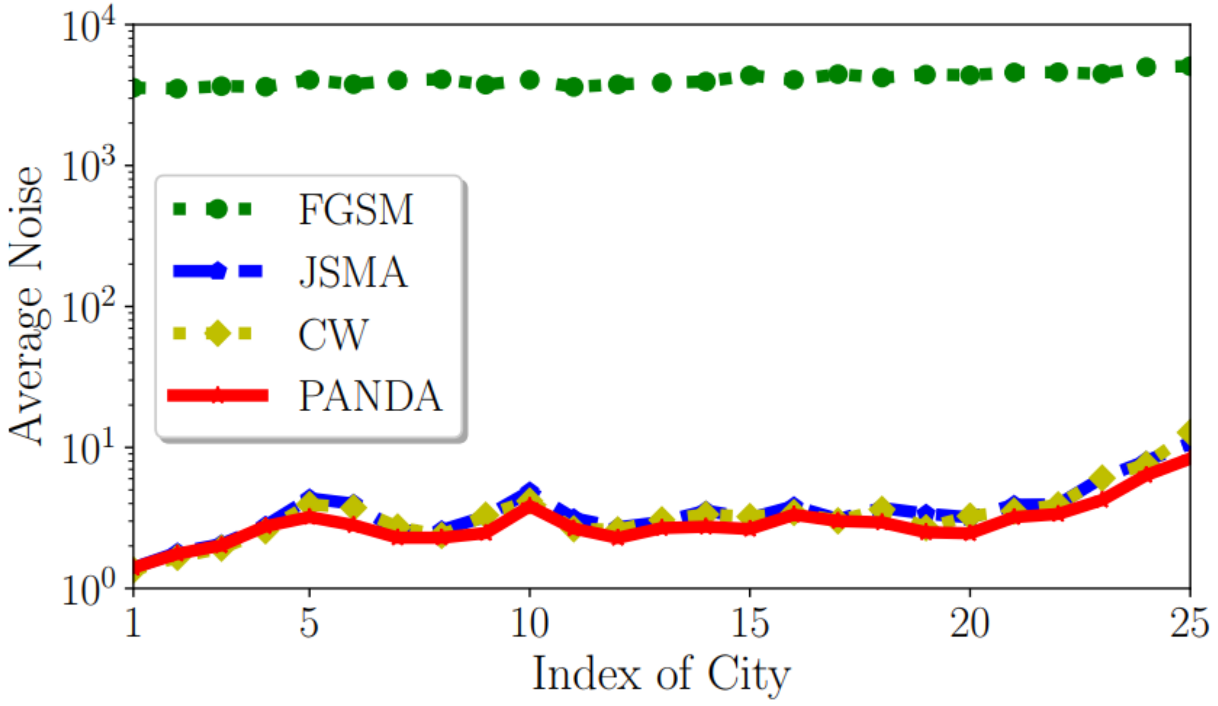
Comparing PANDA with Existing Evasion Attack Methods

- *Fast Gradient Sign Method* (FGSM)

- *Jacobian-based Saliency Map Attack* (JSMA)

- *Carlini and Wagner Attack* (CW)

Average Noise



FGSM adds orders of magnitude larger noise

PANDA adds smaller noise than JSMA

PANDA is comparable to CW

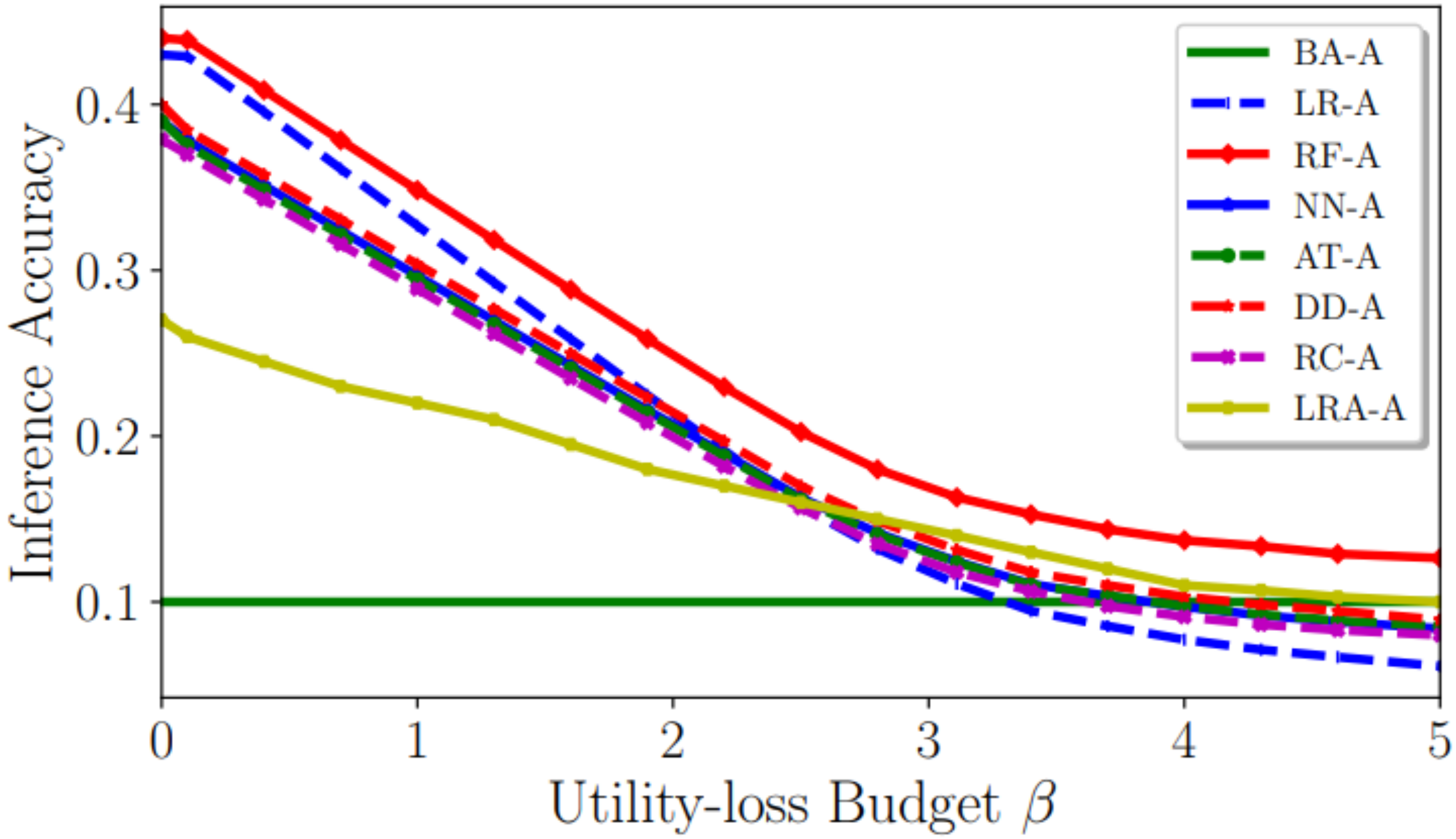
Success Rate and Running Time

Method	Success Rate		Running Time (s)	
	LR-D	NN-D	LR-D	NN-D
FGSM	100%	100%	7.6	84
JSMA	100%	100%	9.0	295
CW	75%	71%	7,406	1,067,610
PANDA	100%	100%	8.7	272

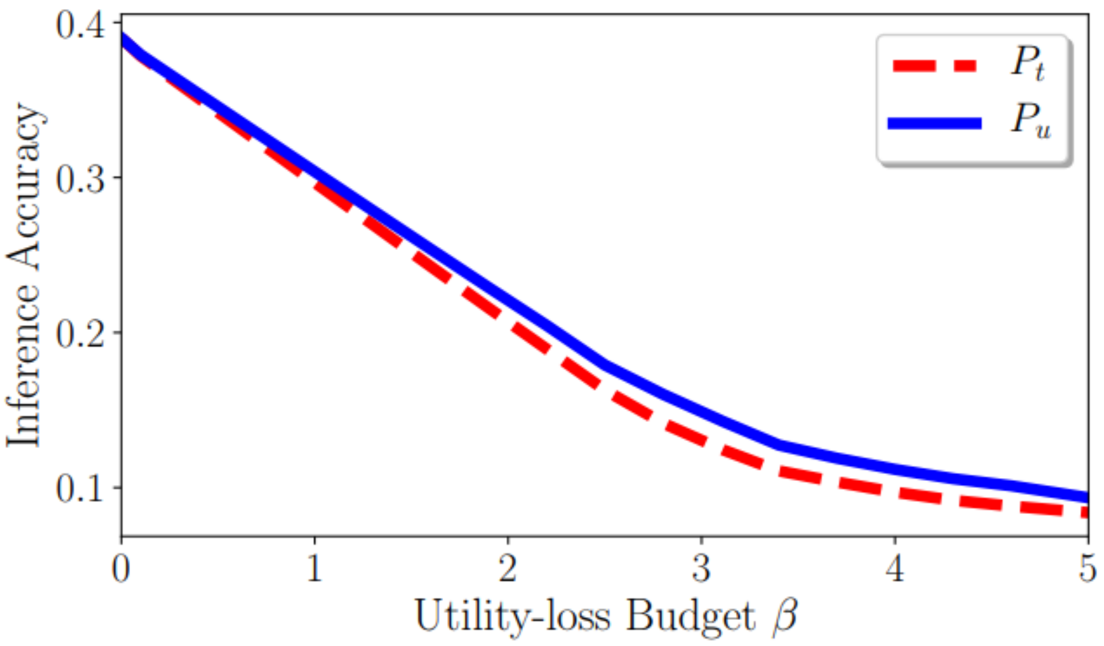
PANDA is slightly faster than JSMA

PANDA is around 800 times and 4,000 times faster than CW for the LR-D and NN-D, respectively

AttriGuard is Effective



Impact of the Target Probability Distribution

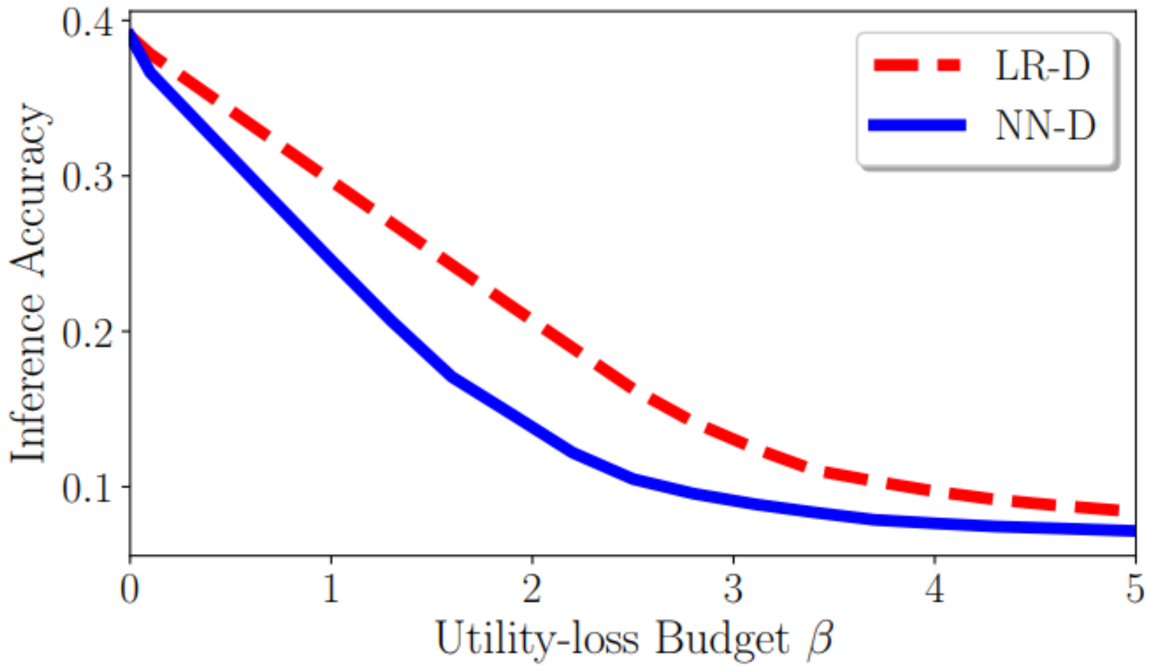


Target probability distribution p_t outperforms p_u

p_t :Estimated target probability distribution using training dataset

p_u :Uniform probability distribution

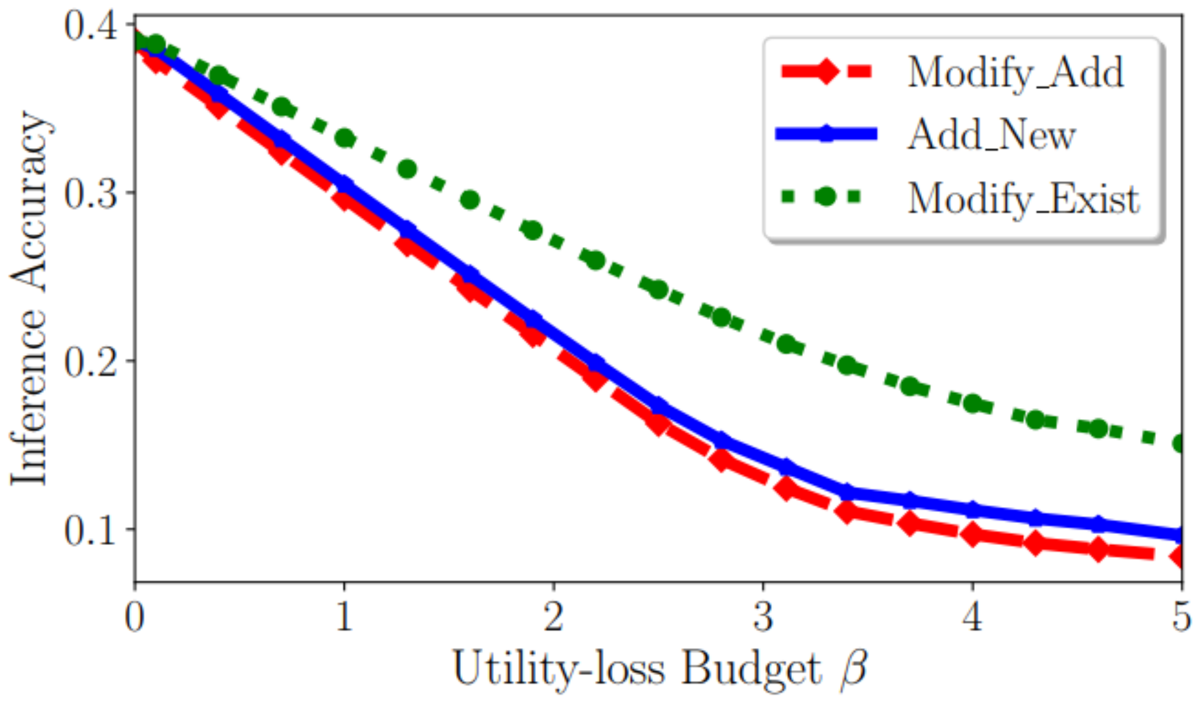
Impact of the Defender's Classifier



Attacker's classifer:
Neural Network(NN-A)

AttriGuard is better when attacker and defender use the same classifier

Impact of Different noise-type-policies



Modify_Add outperforms Add_New, which outperforms Modify_Exist

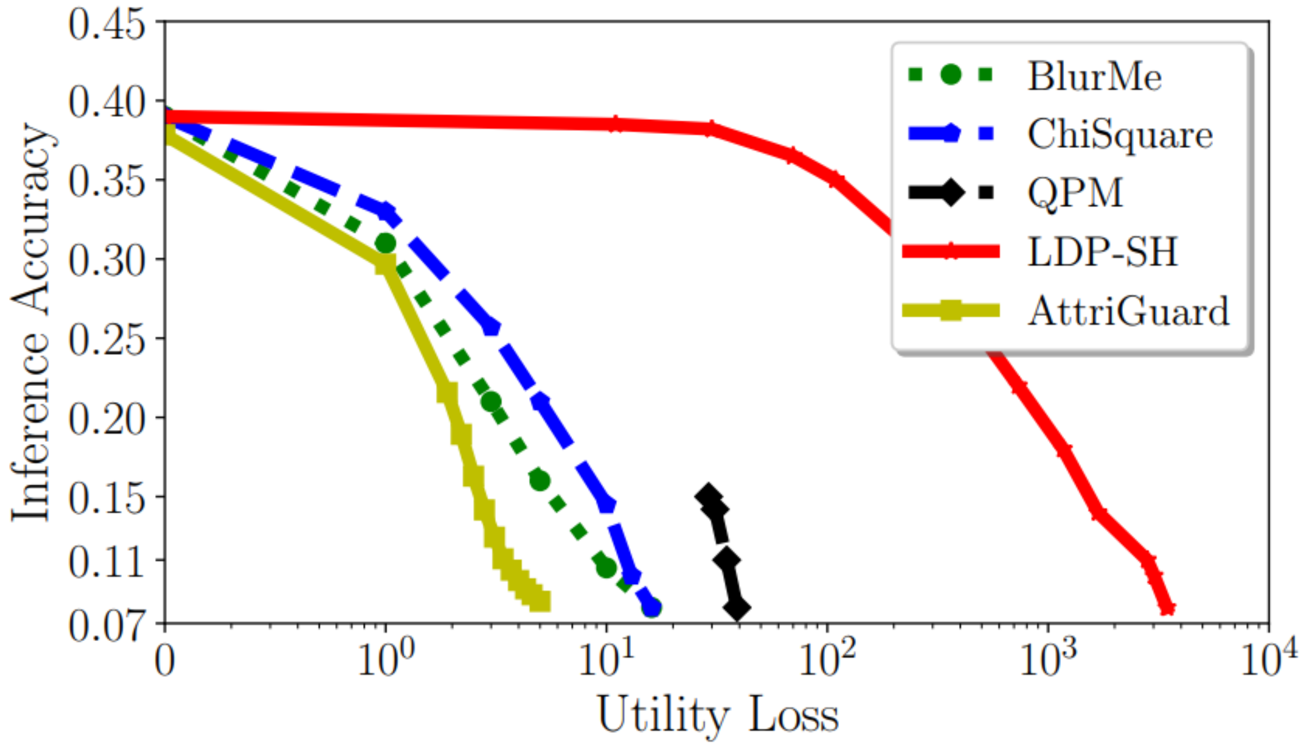
Comparing AttriGuard with Existing Defenses

- Correlation-based Methods
  - BlurMe
  - ChiSquare

- Approximate game-theoretic method
  - Quantization Probabilistic Mapping(QPM)

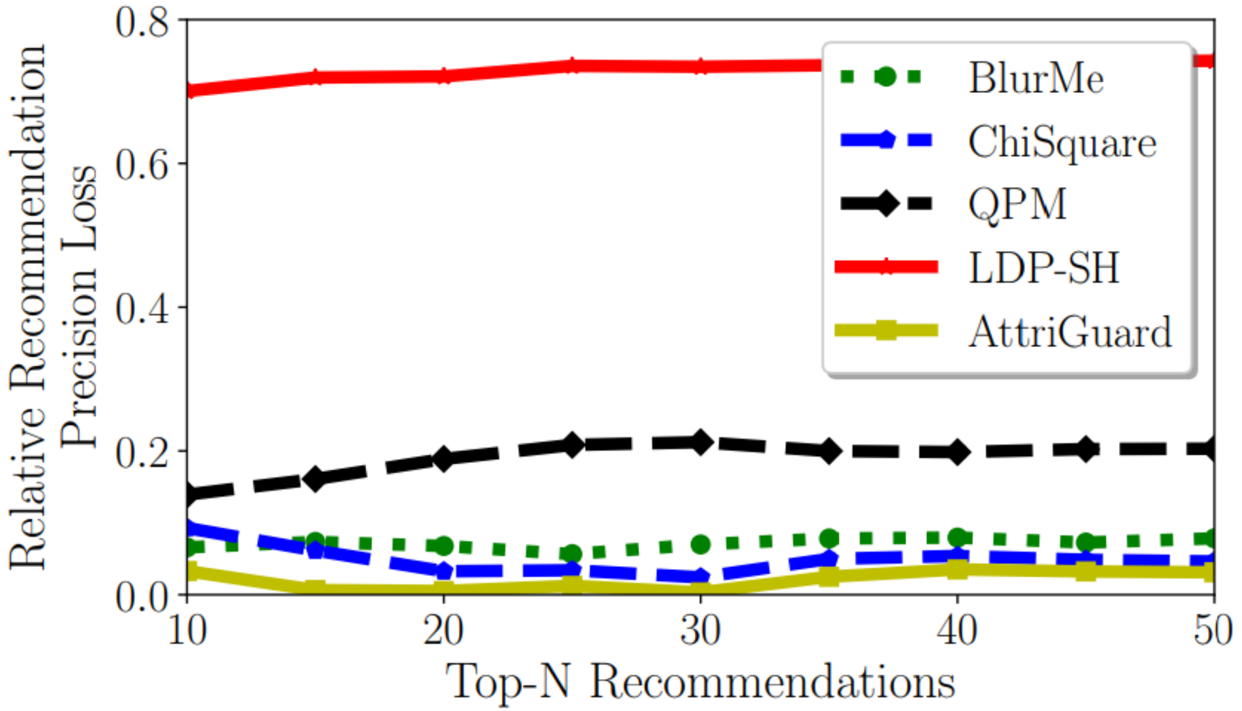
- Local Differential Privacy
  - LDP-SH

Comparing AttriGuard with Existing Defenses



AttriGuard incurs smaller utility-loss

Comparing AttriGuard with Existing Defenses



AttriGuard incurs smaller relative recommendation precision loss

OUTLINE

- Motivation
- Algorithm
- Evaluation
- Conclusion

Conclusion

- AttriGuard can defend against attribute inference attacks with a small utility loss

- Evasion attacks/Adversarial examples can be used as defensive techniques for privacy protection

- AttriGuard significantly outperforms existing defenses