Research Statement

My research interest lies on privacy-preserving machine learning. Currently there're two main problems I want to solve, which are listed below.

Learning private representation.

Data releasing contains risk of privacy leakage. It's useful to know if there's a obfuscation mechanism that could obfuscate the private features while maintains data's utility. Inspired by recent advancement of generative adversarial networks, learning private representation could be formed as a two-player privacy game: a defender that obfuscates data to minimize the risk of private attributes being inferred without losing too much utility, while an active adversary keeps trying to infer sensitive information from the obfuscated data released by defender. The defender and adversary are both assumed neural networks here.

There're already some works on this kind of problems, like Generated Adversarial Privacy(GAP) which uses GAN model to perturbate private information, and applications like DeepPrivacy, which directly blurs human faces for identity. However, the real problenm is to know a fast way to converge the defender's neural network to the optimal setting, or whether there exists such optimal point. Since this is a non-linear optimization problem, we need to define a proper local optimal, which could be determined in a data-driven way.

Similiar works in learning fair or robust representation appears recently. But I think it's worth to look from the privacy prospective.

Privacy-Preserving Federated Learning

Federated learning is proposed by Google to train machine learning models without users uploading the raw data. The server and users train same model and server side receives only gradient updates from user side. However, researchers have proved that gradient itself could release privacy, since the server has access to users' gradient updates. It's proved that in synchronized setting the server could even recover the raw training data if training data are binary(here sycrhonized means client upload gradient change in each epoch), although the server might not know the exact owner of the data. For asynchronized setting, it's unclear whether the server could recover the data with partial infoirmation. Previous works using GAN have claimed success on image datasets.

Currently there're three potential tools that could achieve secure aggregation which prevents this kind of gradient leakage: secure multi-party computation, differential privacy and homomorphic encryption. Each tool has its own advantage and disadvantages. MPC would increase communication cost between

clients. Differential Privacy needs to face privacy-usability tradeoff. Homomorphic Encryption needs heavy computation cost. Single tool may not be enough to solve real life problems, so it's necessary to see how to properly combine them.