UIC COMPUTER SCIENCE

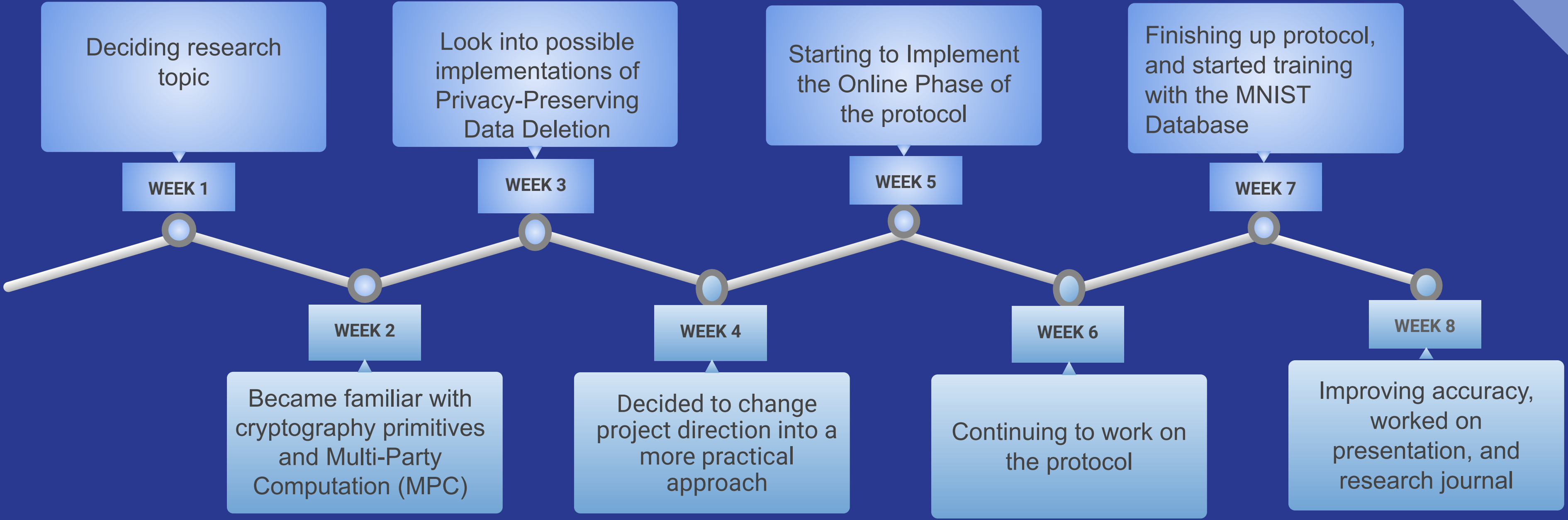
Privacy-Preserving Linear Regression

By Ariadna Fernandez and Sarah Athar
Research Advisor: Dr. Peihan Miao
Graduate Mentor: Shweta Srinivasan

Break Through Tech Research Scholars

BREAK THROUGH TECH CHI

Timeline



WEEK 1

Deciding research topic

WEEK 2

Became familiar with cryptography primitives and Multi-Party Computation (MPC)

WEEK 3

Look into possible implementations of Privacy-Preserving Data Deletion

WEEK 4

Decided to change project direction into a more practical approach

WEEK 5

Starting to Implement the Online Phase of the protocol

WEEK 6

Continuing to work on the protocol

WEEK 7

Finishing up protocol, and started training with the MNIST Database

WEEK 8

Improving accuracy, worked on presentation, and research journal

Privacy-Preserving Machine Learning

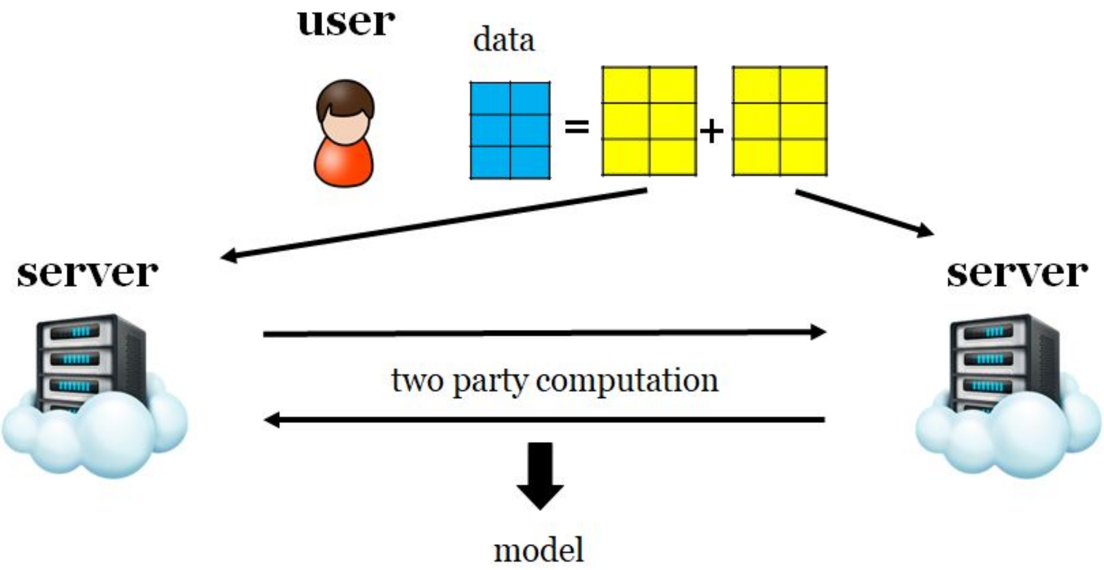
The Problem:

Training Machine Learning (ML) models requires an immense amount of data collection which results in data privacy concerns for data owners that do not want to share sensitive information with other parties.

The Solution:

Privacy Preserving ML provides a solution to this security issue, by enabling companies to perform the same ML algorithm without knowing the underlying content of other parties’ data.

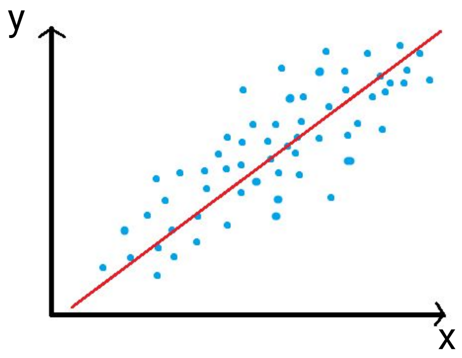
Two-Server Model and MPC



- 2 servers jointly train models, with secret-shared data
- MPC- Multiparty Computation
- Users can go offline after sharing the data, and do not need to interact with the servers during the training

Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data.



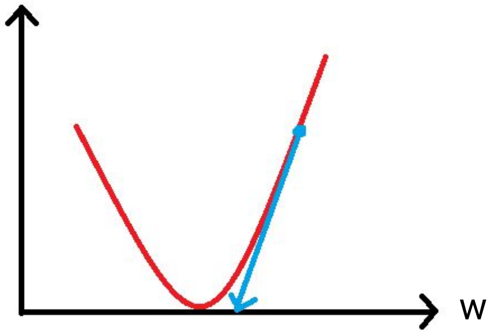
Input: Data Value Pairs - (**x**, y)

Output: Model **w**

$$\boxed{y^*} = \sum_i w_i x_i = \mathbf{w} \cdot \mathbf{x} \approx \boxed{y}$$

Stochastic Gradient Descent (SGD):

1. Initialize **w** randomly
2. Select a random sample (x, y)
3. Update w_i as : $w_i = w_i - \alpha(\mathbf{x} \cdot \mathbf{w} - y)x_i$

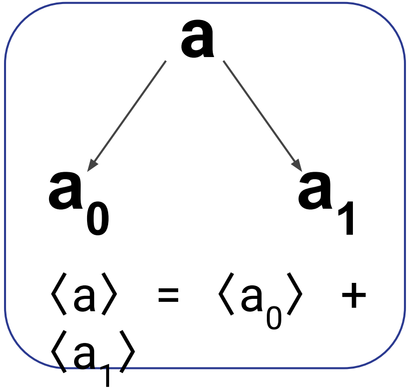


Privacy Preserving Linear Regression

$$\boxed{\text{SGD: } \boldsymbol{w}_i = \boldsymbol{w}_i - \alpha(\boldsymbol{x} \cdot \boldsymbol{w} - \boldsymbol{y})\ \boldsymbol{x}_i}$$

1. Users secret-share and distribute their data samples(x,y) to the two servers.
2. Servers initialize model $\mathbf{w}$ to random values and secret-share it.
3. Servers repeatedly run SGD using pre-computed Multiplication Triplets on the secret-shared values (offline phase).
4. Use truncation techniques on the resulting $\mathbf{w}$ while ensuring accuracy.

Secret-Share Multiplication



$\langle a \rangle = \langle a_0 \rangle + \langle a_1 \rangle$

Secret-sharing is a cryptographic primitive in which a piece of data, or ‘secret’ in this case a, is kept private by having distinct owners hold a share of the secret.

Multiplication triplets: Random Values-$\langle \mathbf{u} \rangle$ $\langle \mathbf{v} \rangle$ $\langle \mathbf{z} \rangle$

To mask data samples $\langle \mathbf{a} \rangle$ and $\langle \mathbf{b} \rangle$ each server computes:

$\langle \mathbf{e} \rangle_i = \langle \mathbf{a} \rangle_i - \langle \mathbf{u} \rangle_i$ $\langle \mathbf{f} \rangle_i = \langle \mathbf{b} \rangle_i - \langle \mathbf{v} \rangle_i$

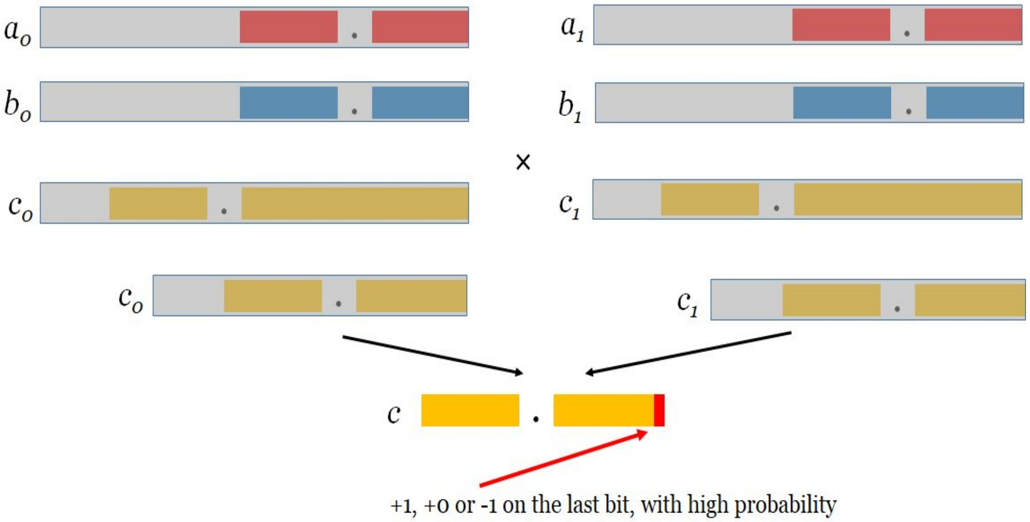
Then exchange their shares of $\langle e \rangle_i$ and $\langle f \rangle_i$:

$\langle \mathbf{e} \rangle = \langle \mathbf{e}_0 \rangle + \langle \mathbf{e}_1 \rangle$ $\langle \mathbf{f} \rangle = \langle \mathbf{f}_0 \rangle + \langle \mathbf{f}_1 \rangle$

To then locally compute:

$\langle \mathbf{c} \rangle_i = -i \cdot \mathbf{e} \cdot \mathbf{f} + \mathbf{f} \cdot \langle \mathbf{a} \rangle_i + \mathbf{e} \cdot \langle \mathbf{b} \rangle_i + \langle \mathbf{z} \rangle_i$

Truncation on Shared Values



- Once Secret Share Multiplication takes place, using the pre-computed triplets, resulting in **c**, this can lead to an overflow.
- Hence to resolve this issue, **c** is truncated independently.

Details on our Implementation (Online Phase):

matrices:

Data features → **X**

Labels → **Y**

Vectorized form of SGD function:

$$\boldsymbol{w} := \boldsymbol{w} - (1|\mathrm{B}|)\ \alpha\ \boldsymbol{X}^{\mathrm{T}}{}_{\mathrm{B}} \times (\mathbf{X}_{\mathbf{B}} \times \boldsymbol{w} - \mathbf{Y}_{\mathrm{B}})$$

- Trained the Protocol on MNIST Database using sample size of .

Offline Phase

- Generate the Necessary Multiplication triplets.
- All communication in the offline phase can be done in one interaction.
- Data Independent.

Online Phase

- Does not involve any cryptographic operations.
- Consists mainly of integer multiplications
- Involves bit shifting, during truncation.
- Trains the model given the data

Future Work

1. Implementing the Offline Phase, which computes the Multiplication Triplets using more advanced cryptographic protocols.
2. Establishing a communication channel between the two servers.
3. Learning rate adjustment.
4. Further testing of performance with MNIST Database.

Special Thanks to
Dr. Peihan Miao, Shweta Srinivasan
and Break Through Tech for the
opportunity!

Q&A!!

UIC COMPUTER SCIENCE

BREAK THROUGH TECH CHI