Towards a Practical Cluster Analysis over Encrypted Data

Jai Hyun Park

Seoul National University (SNU)

Joint work with

Jung Hee Cheon and Duhyeong Kim (SNU)

August 14, 2019, Waterloo, Canada

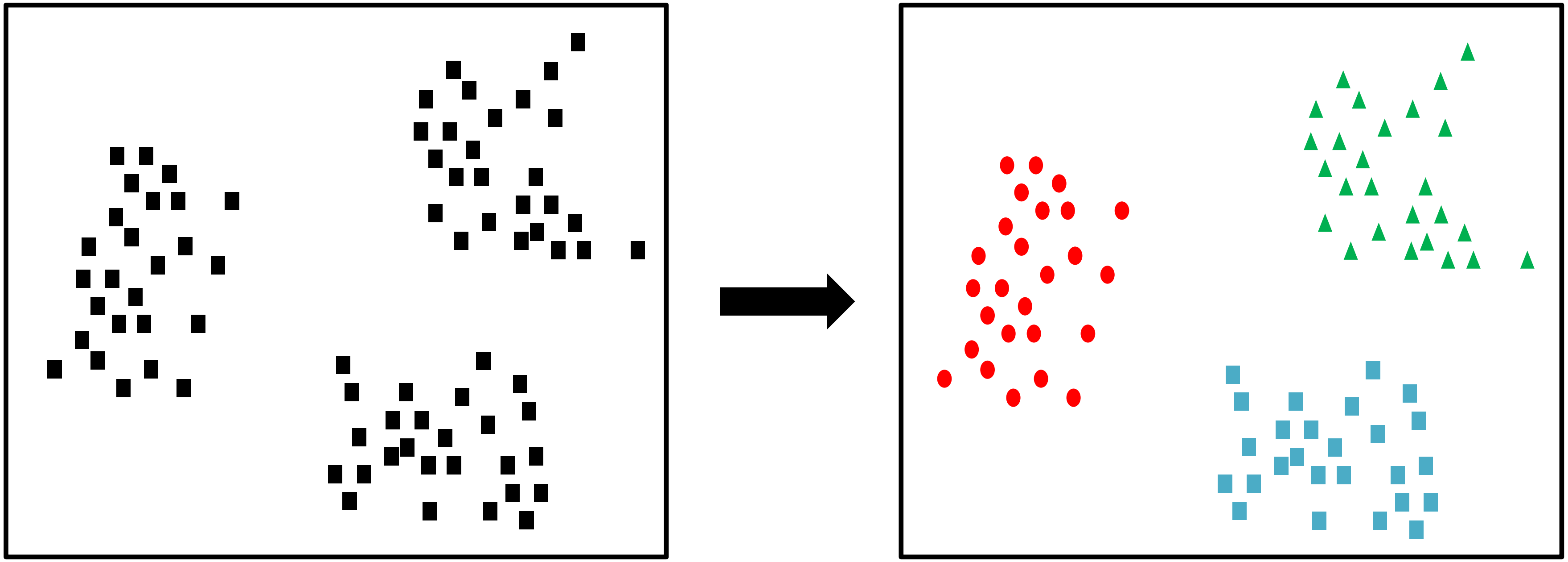
Summary of This Work

- The first privacy preserving non-interactive solution of mean-shift clustering algorithm based on homomorphic encryption

- Outstanding performance: Fast and Accurate
  - 99.99% accuracy on 262,144 data within only 82 min
  - 400 times faster than the previous work (SAC 18)

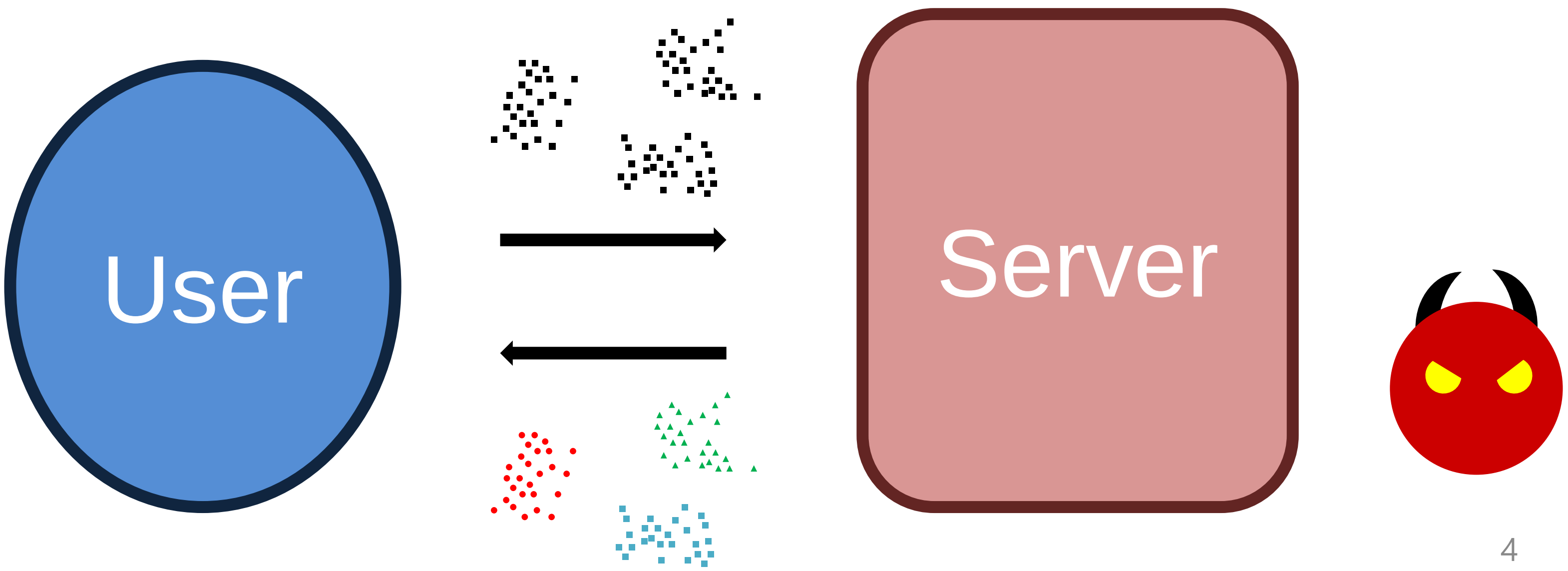
Data Clustering

- Grouping a set of given data into several subgroups
- Unsupervised machine learning task



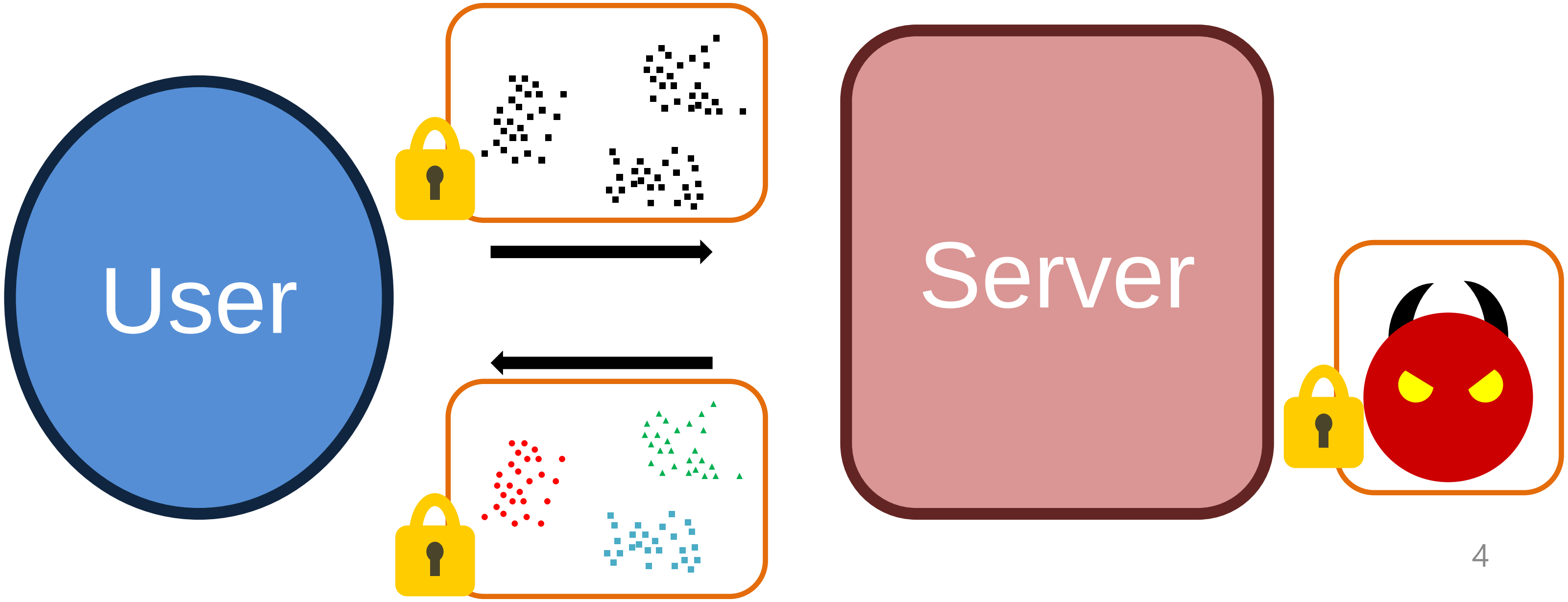
Privacy Preserving Clustering

- Clustering is used in fields dealing with private information
  - Bioinformatics, finance, customer behavior analysis
- People do not want to delegate clustering of raw data to untrusted server



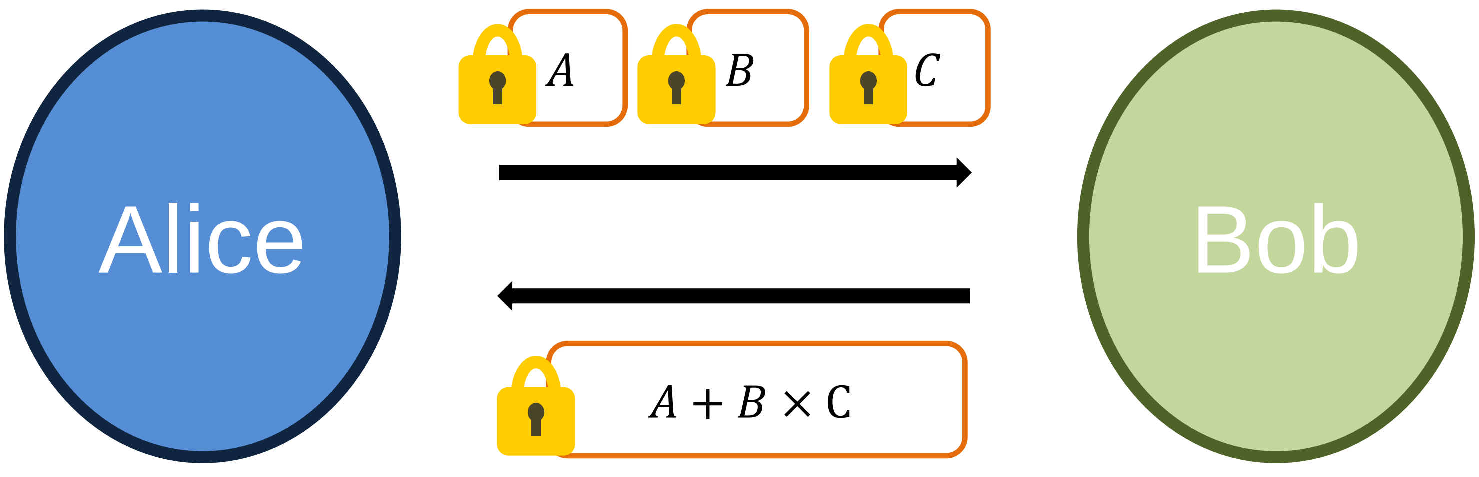
Privacy Preserving Clustering

- Clustering is used in fields dealing with private information
  - Bioinformatics, finance, customer behavior analysis
- People do not want to delegate clustering of raw data to untrusted server



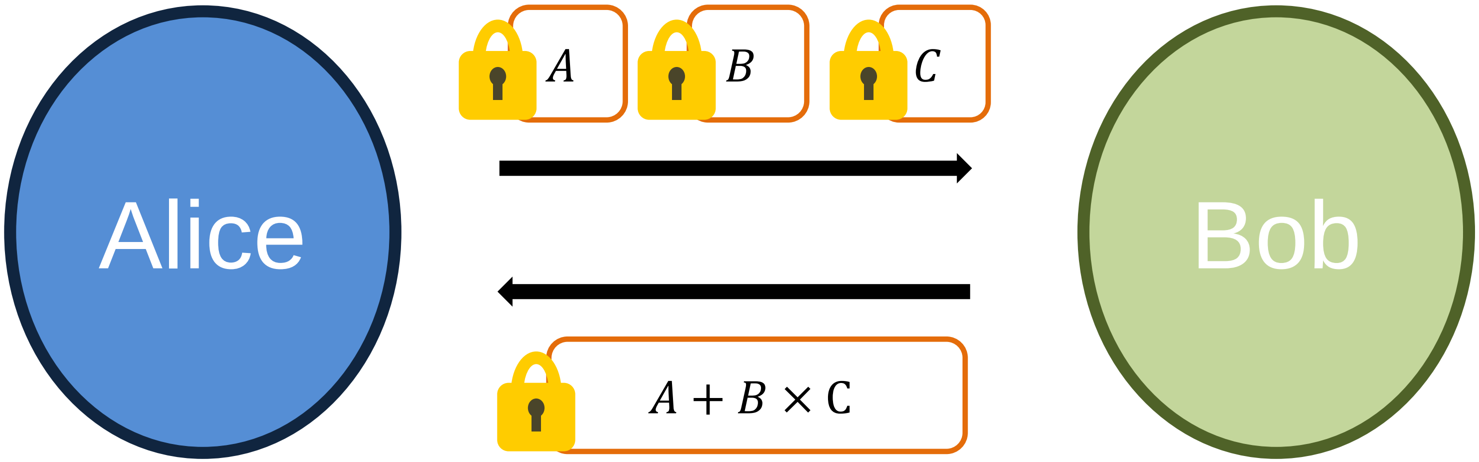
Homomorphic Encryption

- Homomorphic encryption (HE) allows arithmetic (+, -, x) operations on ciphertexts without any decryption process



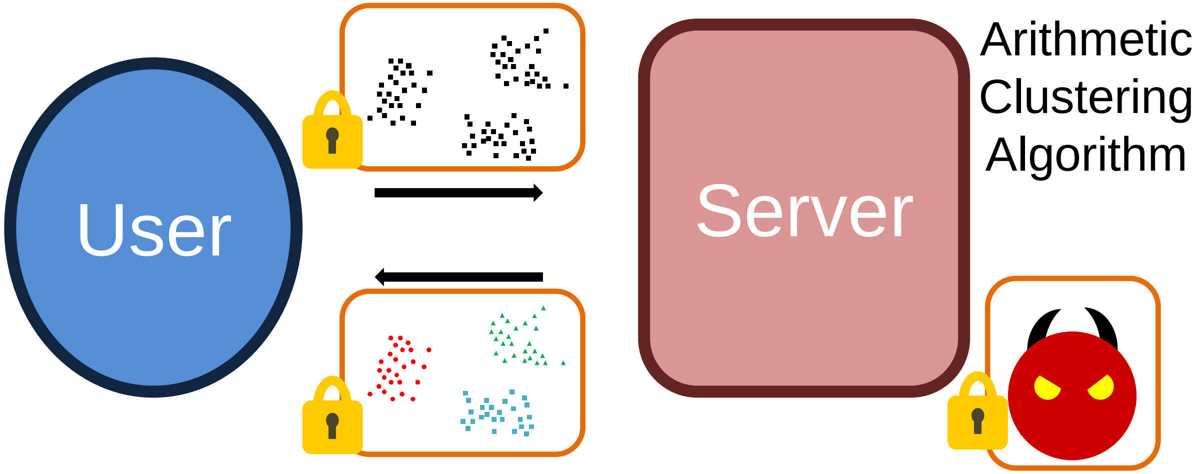
Homomorphic Encryption

- Homomorphic encryption (HE) allows arithmetic (+, -, x) operations on ciphertexts without any decryption process
- Non-arithmetic operations (comparison, min, max) can be approximately computed
  - But expensive



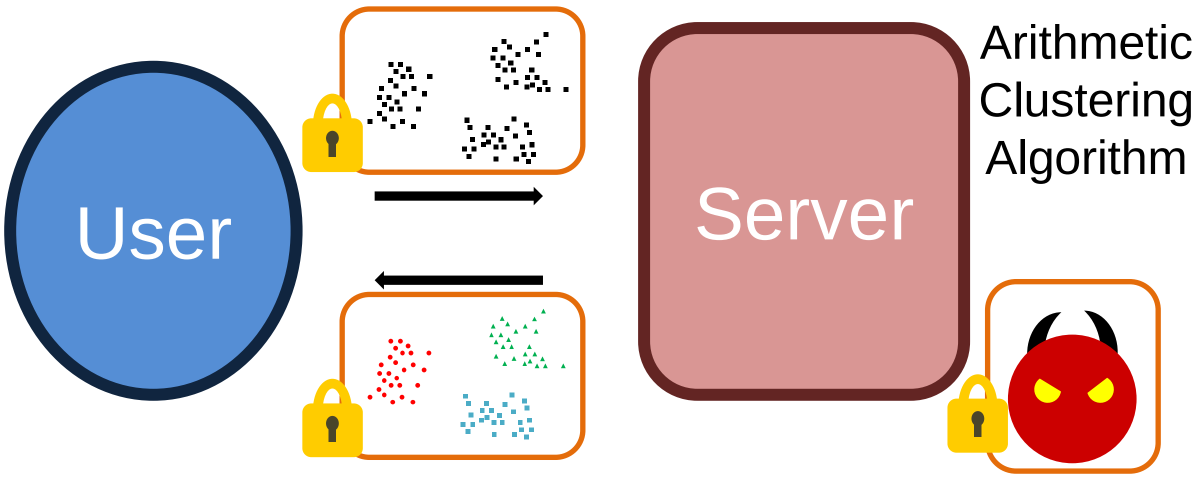
Privacy Preserving Clustering

- People can delegate clustering of private data to untrusted server with homomorphic encryption



Privacy Preserving Clustering

- People can delegate clustering of private data to untrusted server with homomorphic encryption



Two main issues:

1. Which clustering algorithm?
2. How to make it arithmetic?

K-means vs. Mean-shift

- K-means is faster
  - But uses more pieces of information

	K-means Clustering	**Mean-shift Clustering**
Complexity	$O(\#\text{clusters} \cdot \#\text{points} \cdot \#\text{iterations})$	$\mathbf{O}(\#\mathbf{points}^{\mathbf{2}} \cdot \#\mathbf{iterations})$
Parameter	Number of Clusters	**None**
Shape of data	Should be convex	**None**
Comparison Operations	A number of comparison operations	**None**

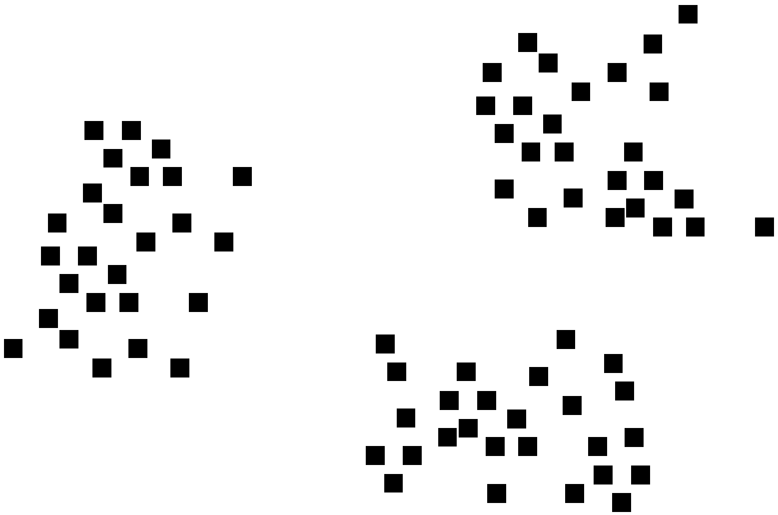
K-means vs. Mean-shift

- K-means is faster
  - But uses more pieces of information
- Mean-shift clustering is more HE applicable
  - Non-parametric
  - No restriction on the shape of data
  - Does not use comparison operations

	K-means Clustering	Mean-shift Clustering
Complexity	$O(\#\text{clusters} \cdot \#\text{points} \cdot \#\text{iterations})$	$\mathbf{O(\#points^2 \cdot \#iterations)}$
Parameter	Number of Clusters	**None**
Shape of data	Should be convex	**None**
Comparison Operations	A number of comparison operations	**None**

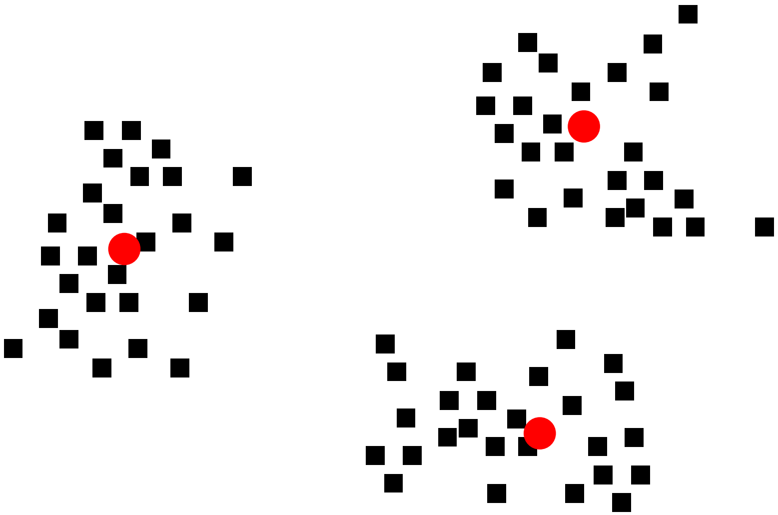
Mean-shift Clustering

- Clustering technique based on an estimated density map
  - Label each point by its closest local maximum (mode) of a Kernel Density Estimator (KDE)



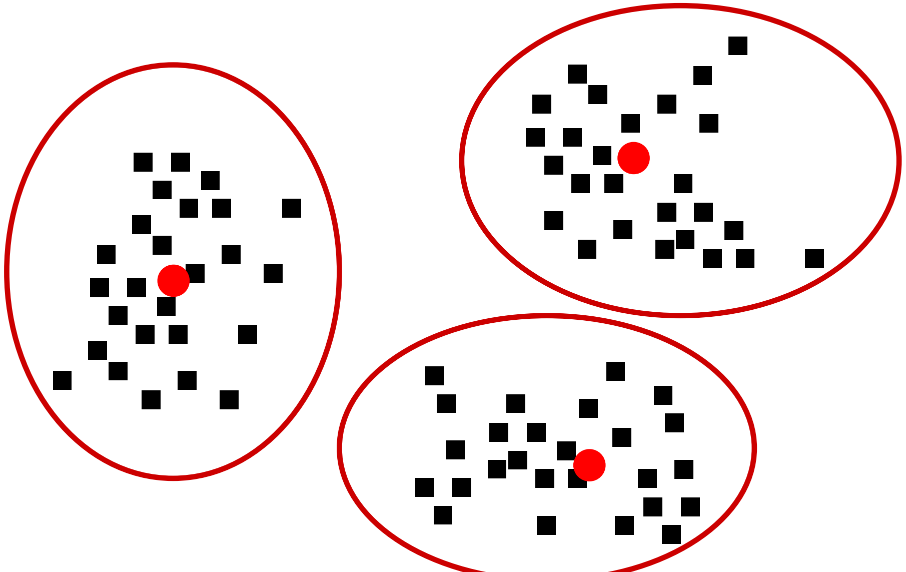
Mean-shift Clustering

- Clustering technique based on an estimated density map
  - Label each point by its closest local maximum (mode) of a Kernel Density Estimator (KDE)



Mean-shift Clustering

- Clustering technique based on an estimated <u>density map</u>
  - Label each point by its closest local maximum (mode) of a Kernel Density Estimator (KDE)



Mean-shift Clustering

- **Kernel function**
  - A function indicating a probability density map generated by a given datum

Mean-shift Clustering

$$K(x, P_i) = c_k k(\|P_i - x\|^2)$$

profile k is a non-negative and decreasing function

- **Kernel function**
  - A function indicating a probability density map generated by a given datum

Mean-shift Clustering

- **Kernel function**

$$K(x, P_i) = c_k k(\|P_i - x\|^2)$$

profile k is a non-negative and decreasing function

  - A function indicating a probability density map generated by a given datum

- **KDE map**

$$F(\boldsymbol{x}) = \frac{1}{p} \cdot \sum_{i=1}^{p} K(\boldsymbol{x}, P_i)$$

  - Estimator of probability density function based on the given kernel function

Mean-shift Clustering

- **Kernel function**

$$K(x, P_i) = c_k k(\|P_i - x\|^2)$$

profile k is a non-negative and decreasing function

  – A function indicating a probability density map generated by a given datum

- **KDE map**

$$F(\boldsymbol{x}) = \frac{1}{p} \cdot \sum_{i=1}^{p} K(\boldsymbol{x}, P_i)$$

  – Estimator of probability density function based on the given kernel function

- **Modes**
  – The local maxima of the KDE map

Mean-shift Clustering

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \left(\sum_{i=1}^{p} \frac{k'(||\boldsymbol{x} - P_i||^2)}{\sum_{j=1}^{p} k'(||\boldsymbol{x} - P_j||^2)} \cdot P_i - \boldsymbol{x} \right)$$

- **Mean-shift process**
  - Slightly moves each $\boldsymbol{x}$ to a denser point
  - Gradient descent method to seek modes

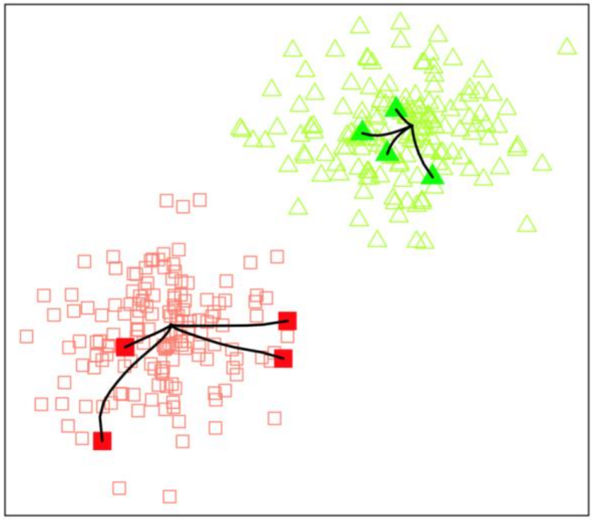
Mean-shift Clustering

- **Mean-shift process**

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \left(\sum_{i=1}^{p} \frac{k'(||\boldsymbol{x} - P_i||^2)}{\sum_{j=1}^{p} k'(||\boldsymbol{x} - P_j||^2)} \cdot P_i - \boldsymbol{x} \right)$$

  - Slightly moves each x to a denser point
  - Gradient descent method to seek modes

- **Mean-shift clustering**
  - Cluster each point by the mode it goes by mean-shift processes



Drawbacks of Mean-shift

1. Non-arithmetic kernel function

– Gaussian kernel function

  - $K_G(x,y) = c_{k_G} \cdot e^{-\frac{\|x-y\|^2}{\sigma^2}}$
  - Exponential function

2. Computationally expensive

– $O(\#\mathrm{points}^2 \cdot \#\mathrm{iterations})$

IDEA1: HE Friendly Kernel

- **<u>New kernel function</u>**

$$k(x) = (1 - x)^{2^{\Gamma}+1}$$

1. Similar performance with usual kernels
   - Satisfies the necessary conditions of kernel functions
     - Decreasing and non-negative on its domain
   - Manage to group plaintexts of public datasets properly

IDEA1: HE Friendly Kernel

- **New kernel function**

$$k(x) = (1 - x)^{2^{\Gamma}+1}$$

※ $k_g(x) = e^{-\frac{x}{\sigma^2}}$

1. Similar performance with usual kernels
   - Satisfies the necessary conditions of kernel functions
     - Decreasing and non-negative on its domain
   - Manage to group plaintexts of public datasets properly

IDEA1: HE Friendly Kernel

- **<u>New kernel function</u>**

$$k(x) = (1-x)^{2^{\Gamma}+1}$$

※ $k_g(x) = e^{-\frac{x}{\sigma^2}}$

1. Similar performance with usual kernels
   - Satisfies the necessary conditions of kernel functions
     - Decreasing and non-negative on its domain
   - Manage to group plaintexts of public datasets properly
2. Arithmetic
3. Efficient
   - Requires log degree number of computations

IDEA2: Dust Sampling Method

- Shift only sampled points (dusts) rather than all points

IDEA2: Dust Sampling Method

- Shift only sampled points (dusts) rather than all points
  - $O(\#\mathrm{dusts} \cdot \#\mathrm{points}) < O(\#\mathrm{points}^2)$

IDEA2: Dust Sampling Method

- Shift only sampled points (dusts) rather than all points
  - $O(\#\text{dusts} \cdot \#\text{points}) < O(\#\text{points}^2)$
  - Cannot label all points only by mean-shift process on sampled dusts

IDEA2: Dust Sampling Method

- Shift only sampled points (dusts) rather than all points
  - $O(\#\mathrm{dusts} \cdot \#\mathrm{points}) < O(\#\mathrm{points}^2)$
  - Cannot label all points only by mean-shift process on sampled dusts
  - But, can seek modes of KDE

IDEA2: Dust Sampling Method

- Shift only sampled points (dusts) rather than all points
  - $O(\#\text{dusts} \cdot \#\text{points}) < O(\#\text{points}^2)$
  - Cannot label all points only by mean-shift process on sampled dusts
  - But, can seek modes of KDE

- Label each point by its closest mode
  - $O(\#\text{dusts} \cdot \#\text{points})$

IDEA2: Dust Sampling Method

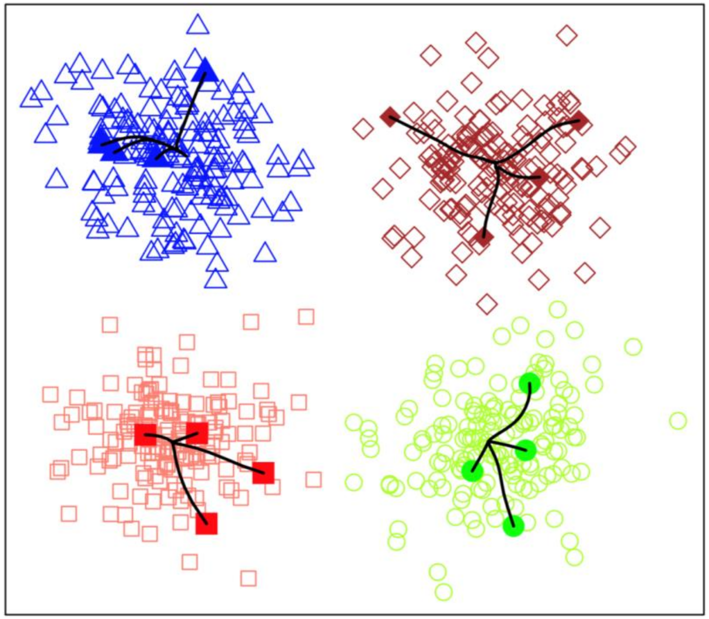
	Original Mean-shift	Dust Sampling Method
Mean-shift	All points	**<u>Only sampled points</u>**
Structure	Find the modes and label the points at the same time	**Find the modes first, and label the points later**
Computational Complexity	$O(\#points^2)$	$\underline{\mathbf{O(\#dusts \cdot \#points)}}$

Our Modified Scheme

1. Sample *dusts* from given data

2. Apply mean-shift to dusts and find modes
   – Use HE friendly kernel

3. Label each points to its closest mode



Experimental Result

- **High accuracy on public datasets**
  - Covers various features of dataset: shape of data, number of data, number of attributes, and number of clusters
- **Fast and accurate performance on large scale dataset**

	Num of Data	Num of Attributes	Num of Clusters	Comp. Time	Quality Evaluation	
					Accuracy	Silh Coeff
Hepta	212	3	7	25 min	212/212	0.702 (0.702)
Tetra	400	3	4	36 min	400/400	0.504 (0.504)
Two Diamonds	800	2	2	38 min	792/800	0.478 (0.485)
Large Scale	262,144	4	4	82 min	262127 /262144	0.781 (0.781)

※ Use multi-threading (8 threads)

Experimental Result

- 400 times faster than the previous work (JA18) on Lsun public dataset

	JA18	Our work
Comp. Time	25.79 days	**83 min**
HE library	TFHE	**HEAAN**

※ Use a single thread

[JA18] Jäschke, A. and Armknecht, F., 2018, August. Unsupervised machine learning on encrypted data. In International Conference on Selected Areas in Cryptography (pp. 453-478). Springer, Cham.

Q&A
Thank you!