Large-scale Wireless Fingerprints Prediction for Cellular Network Positioning

Xinyu Wu, Xiaohua Tian, Xinbing Wang

Shanghai Jiao Tong University

April 18, 2018

Outline

Fingerprint **Prediction** & **Positioning**

- Introduction & Motivation
- Problem Formulation
- Algorithm Design
- Experimental Validation
- Conclusion

Outline

- **Introduction & Motivation**
- Problem Formulation
- Algorithm Design
- Experimental Validation
- Conclusion

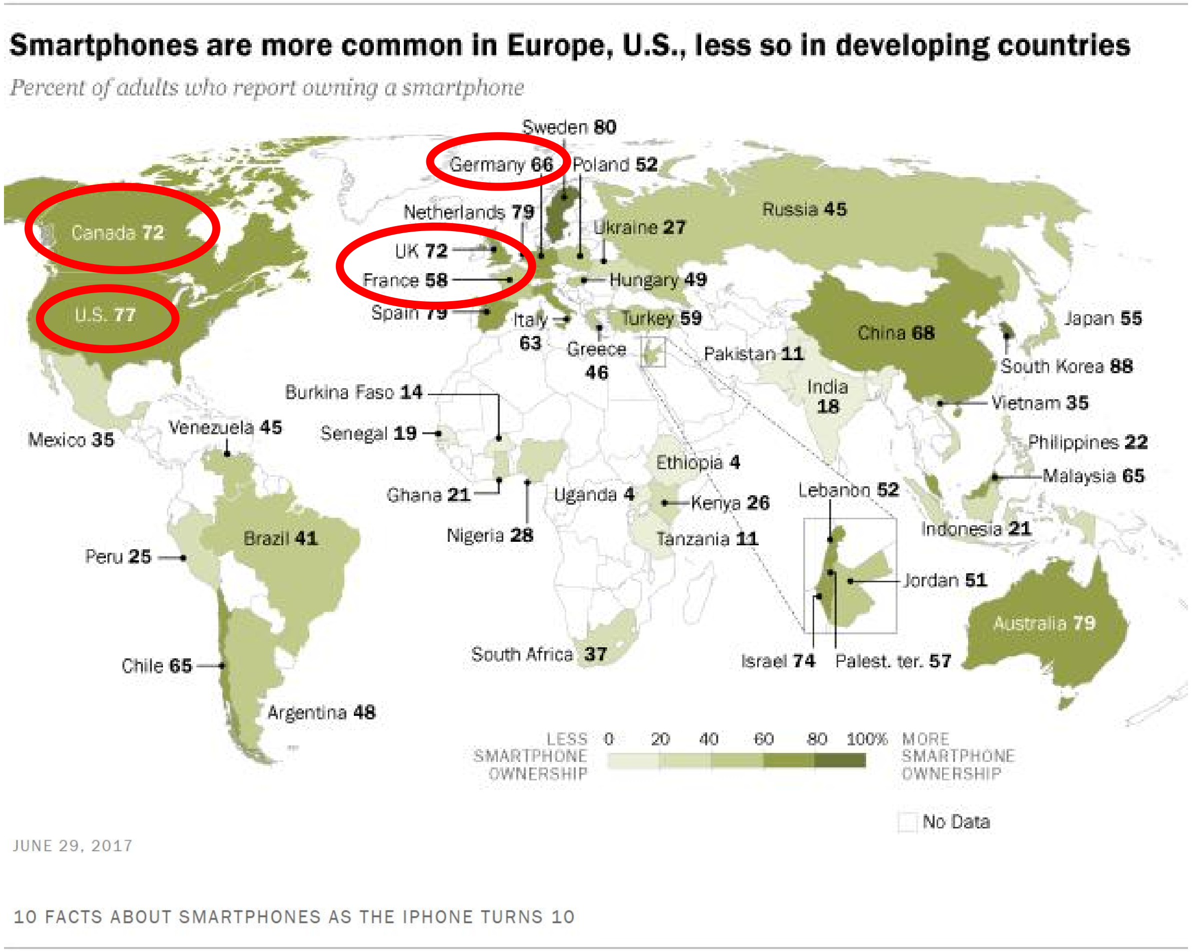
Introduction

Introduction

911 Requirement: Localizing emergency caller

Challenge 1: Weak GPS Signal

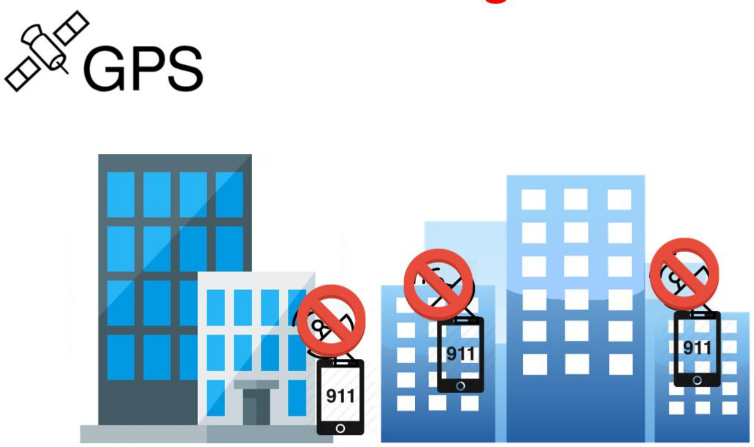
Challenge 2: Device Constraint



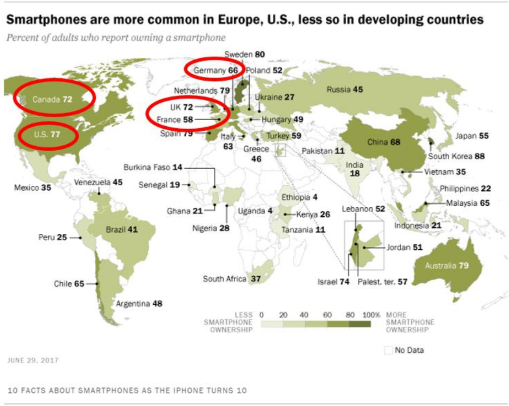
Introduction

Alternative: Fingerprinting localization

Weak GPS Signal

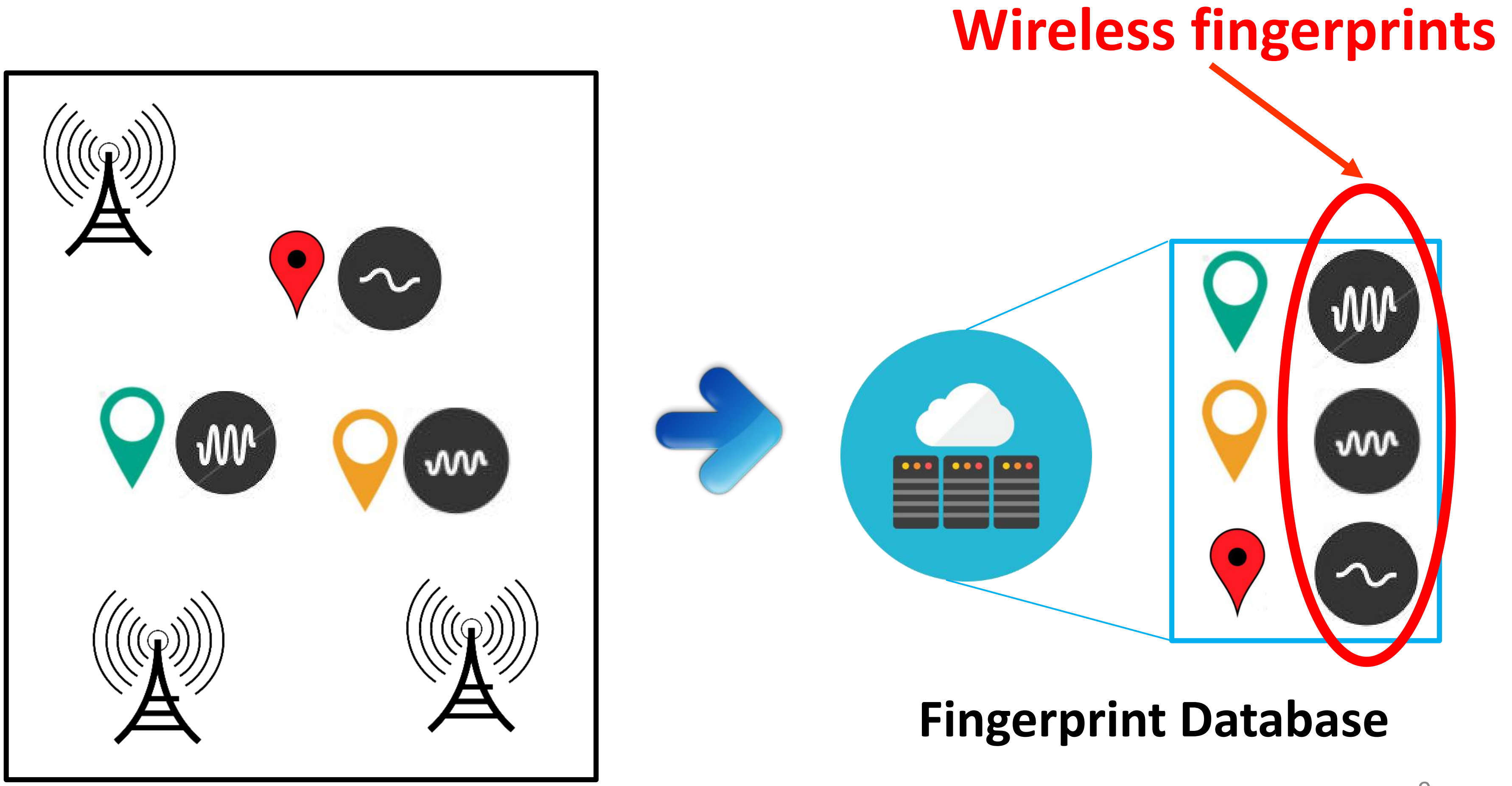


Cellphones as basic devices



Fingerprinting Localization

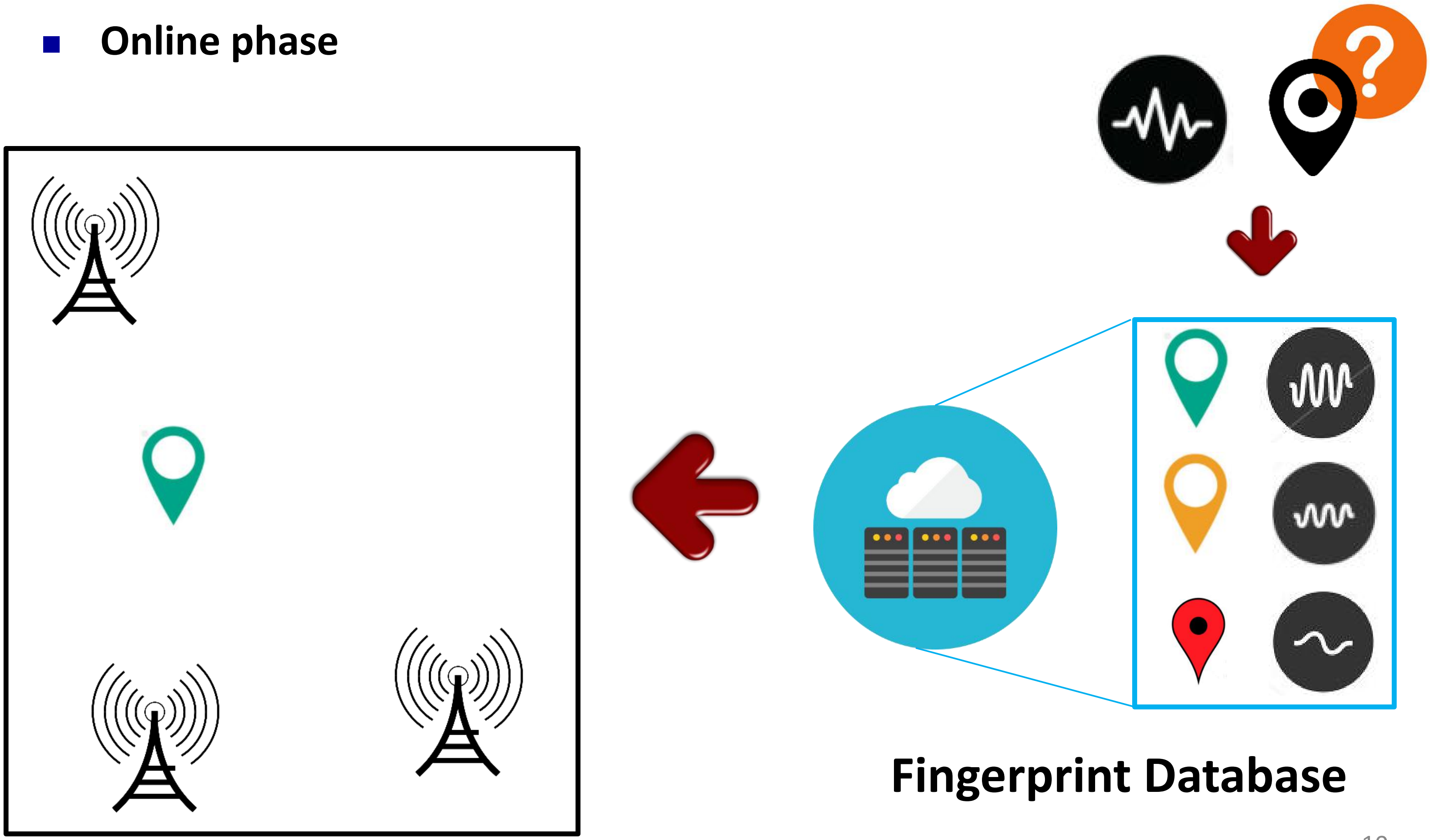
- **Method:** **Offline phase + Online phase**
  - **Offline phase**



Fingerprinting Localization

❖ **Method: Offline phase + Online phase**

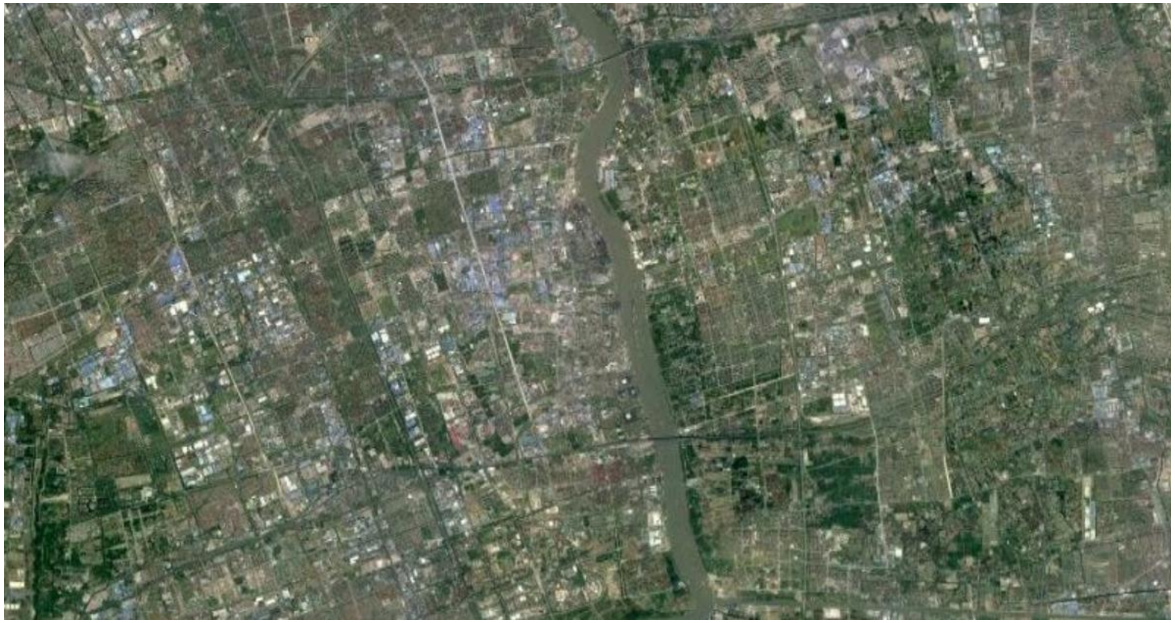
- **Online phase**



Challenge for Fingerprinting Localization

- **Constructing Fingerprint Database**

Large Area Outdoors



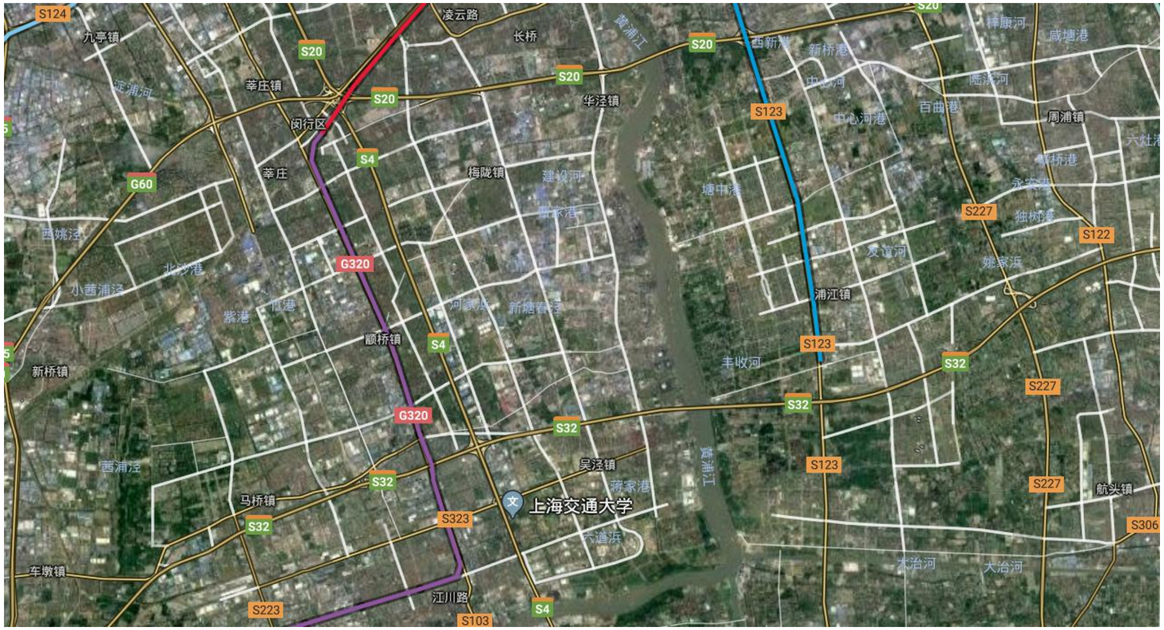
Challenge: Requiring a large number of sampled fingerprints

Costly !

Challenge for Fingerprinting Localization

- **Constructing Fingerprint Database**

Large Area Outdoors



Idea: Sample easily available fingerprints, and predict others

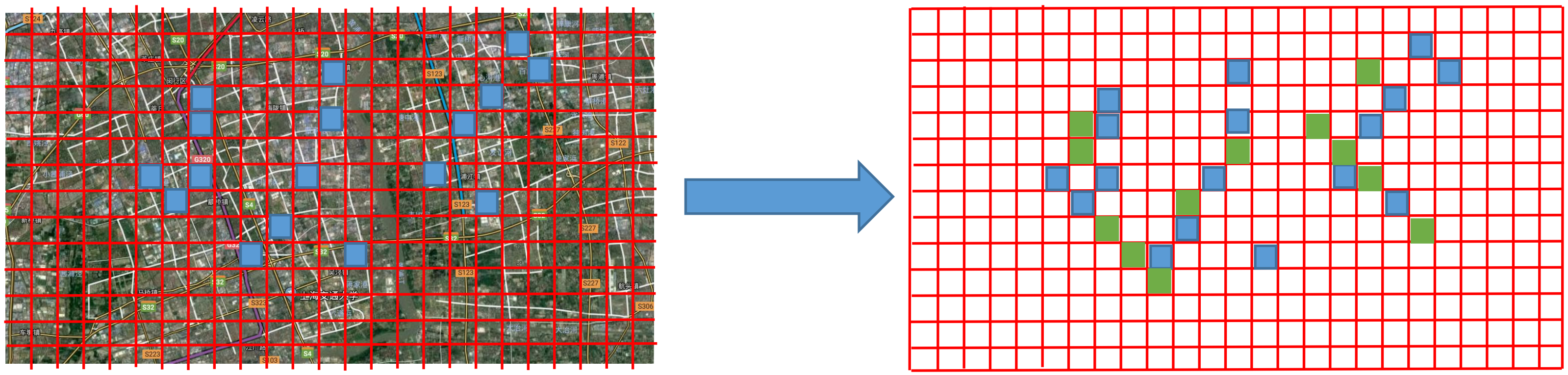
Fingerprint Prediction

Outline

- Introduction & Motivation
- **Problem Formulation**
- Algorithm Design
- Experimental Validation
- Conclusion

Problem Formulation

- Modeling Fingerprint Prediction
  - Grid the target area
  - Have some sampled areas
  - Predict unsampled areas
    - Fingerprints do NOT change dramatically.
    - Fingerprints have correlation.



Target Area

Sampled Fingerprint Value

Predicted Fingerprint Value

Problem Formulation

- Fingerprint Prediction -> Matrix Completion

-51	?	?
?	-70	-57
-68	?	-59

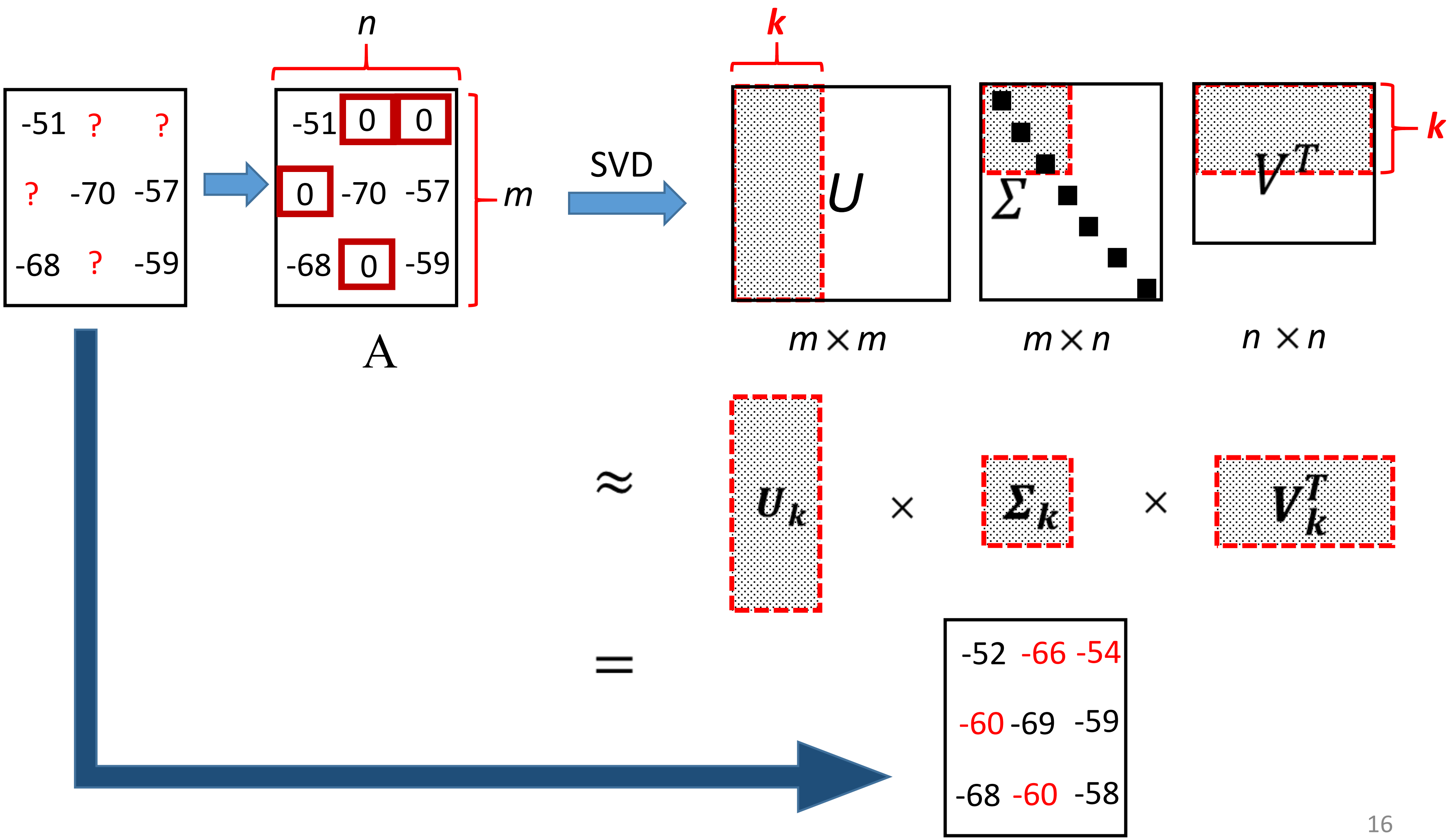
Predict "?"s

-51	-45	-66
-74	-70	-57
-68	-63	-59

SVD usually used to do the prediction

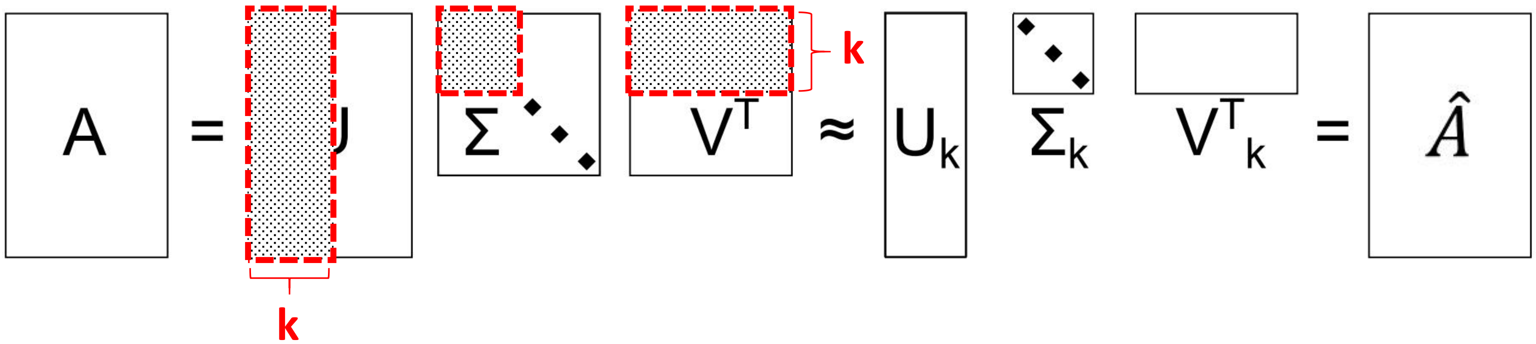
Problem Formulation

- Singular Value Decomposition (SVD)



Problem Formulation

- Singular Value Decomposition (SVD)



-51	?	?
?	-70	-57
-68	?	-59

SVD

k=2

-52	-66	-54
-60	-69	-59
-68	-60	-58

Estimation deviates from the sampled value

Problem Formulation

❖ Matrix Completion: **Minimizing the deviation**

A :

-51	?	?
?	-70	-57
-68	?	-59

$\hat{A}$:

-52	-66	-54
-60	-69	-59
-68	-60	-58

$$\min ||P(A) - P(\hat{A})||$$

$P(A)$:

-51		
	-70	-57
-68		-59

$P(\hat{A})$:

-52		
	-69	-59
-68		-58

Problem Formulation

- Different fillings of *A* generate different results.

-51	?	?
?	-70	-57
-68	?	-59

-51	0	0
0	-70	-57
-68	0	-59

Filling 1 $\xrightarrow{\text{SVD}}$ $||P(A) - P(\hat{A})||$

-51	-1	-1
-62	-70	-57
-68	-90	-59

Filling 2 $\xrightarrow{\text{SVD}}$ $||P(A) - P(\hat{A})||$

-51	-53	-62
-72	-70	-57
-68	-90	-59

Filling 3 $\xrightarrow{\text{SVD}}$ $||P(A) - P(\hat{A})||$

-51	-73	-62
-72	-70	-57
-68	-90	-59

Filling 4 $\xrightarrow{\text{SVD}}$ $||P(A) - P(\hat{A})||$

……

…… ……

Filling N $\xrightarrow{\text{SVD}}$ $||P(A) - P(\hat{A})||$

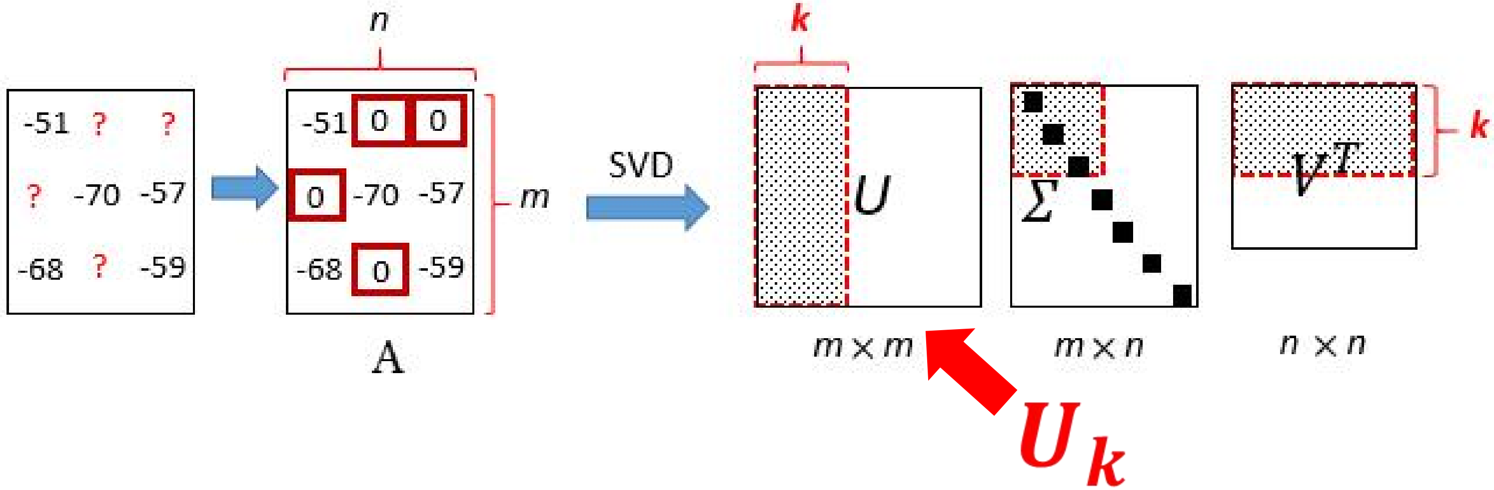
Try all possibilities? Too costly!

Outline

- Introduction & Motivation
- Problem Formulation
- **Algorithm Design**
- Experimental Validation
- Conclusion

Deeper Insight of SVD

- Recall SVD



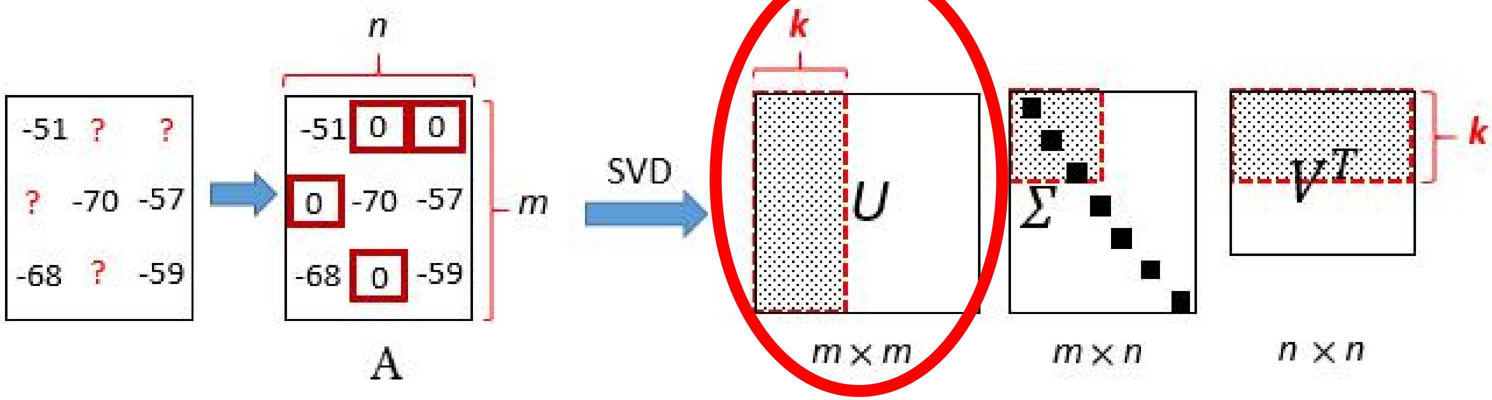
$$||\boldsymbol{P}(\boldsymbol{A}) - \boldsymbol{P}(\widehat{\boldsymbol{A}})|| = f(U_k)$$

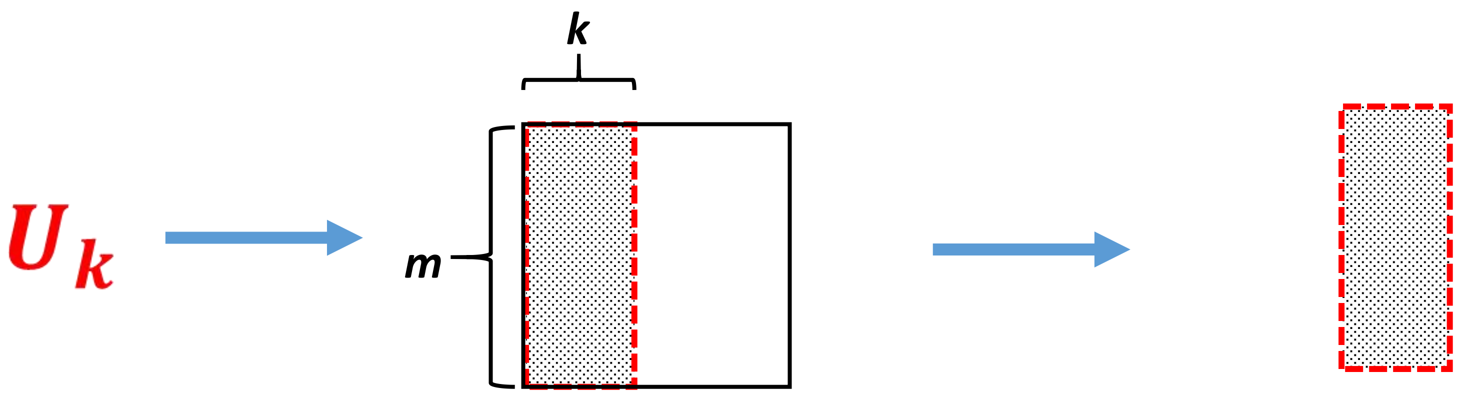
$$\boxed{\min: f(U_k)}$$

Best U_k leading to best estimation！

Deeper Insight of SVD

❖ **What is U_k?**

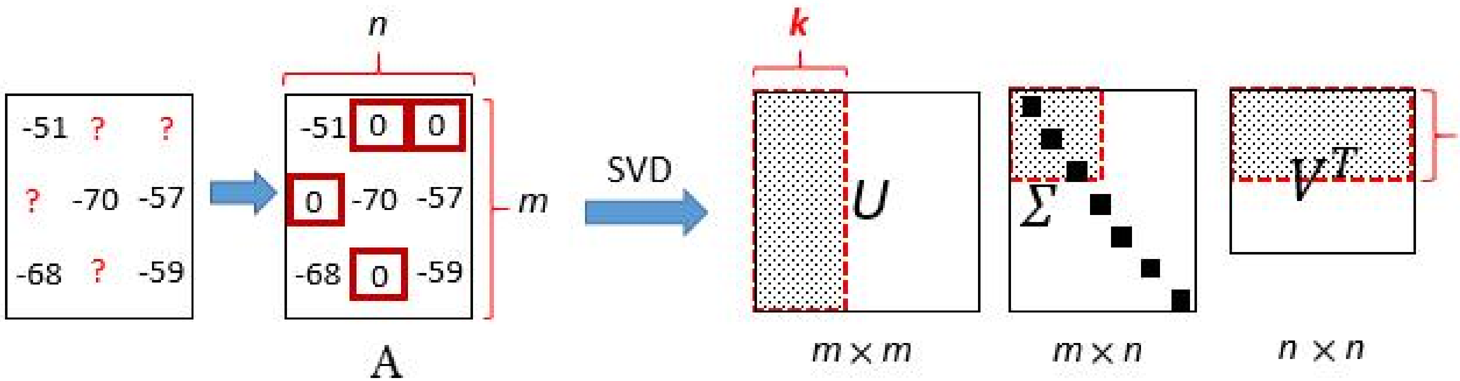




Part of the unitary matrix U **k orthogonal vectors in m-D space**

Deeper Insight of SVD

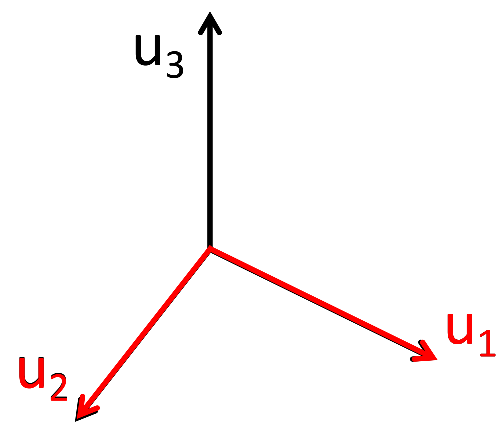
❖ What is U_k?



Example:

m=3
n=3

$$U = \begin{array}{ccc} u_1 & u_2 & u_3 \end{array}$$
$$U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



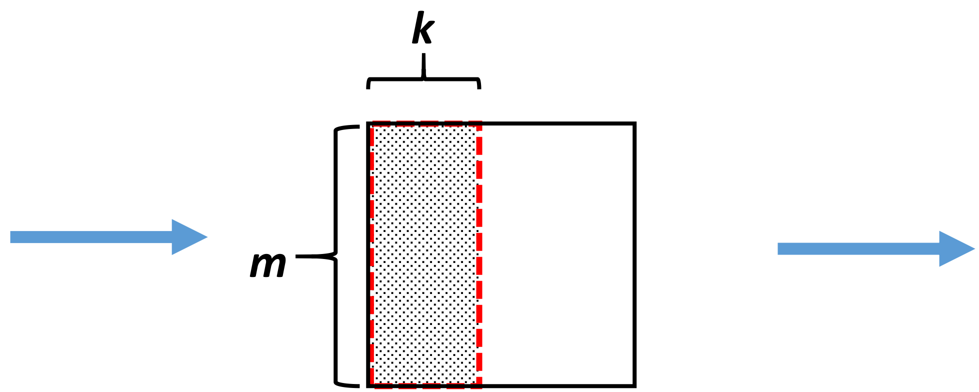
k=2

$$U_k = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 0 \end{pmatrix}$$

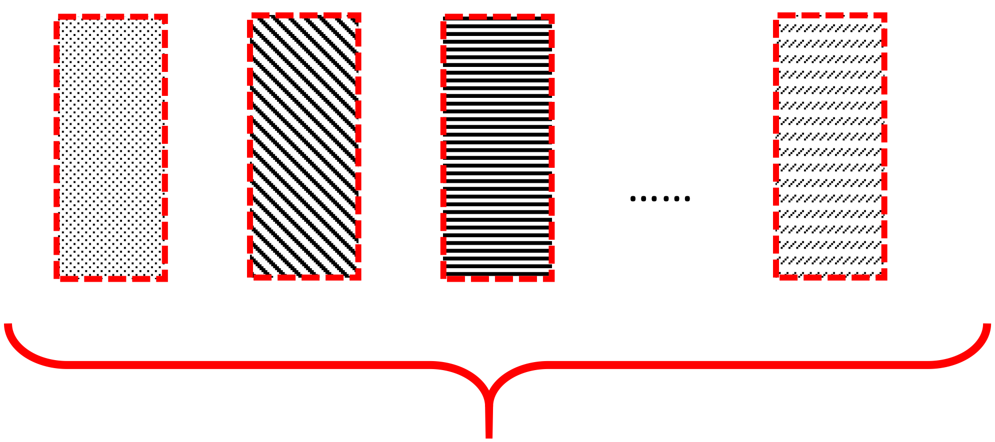
Stiefel Manifold

❖ **Stiefel manifold**

$\boldsymbol{U_k}$



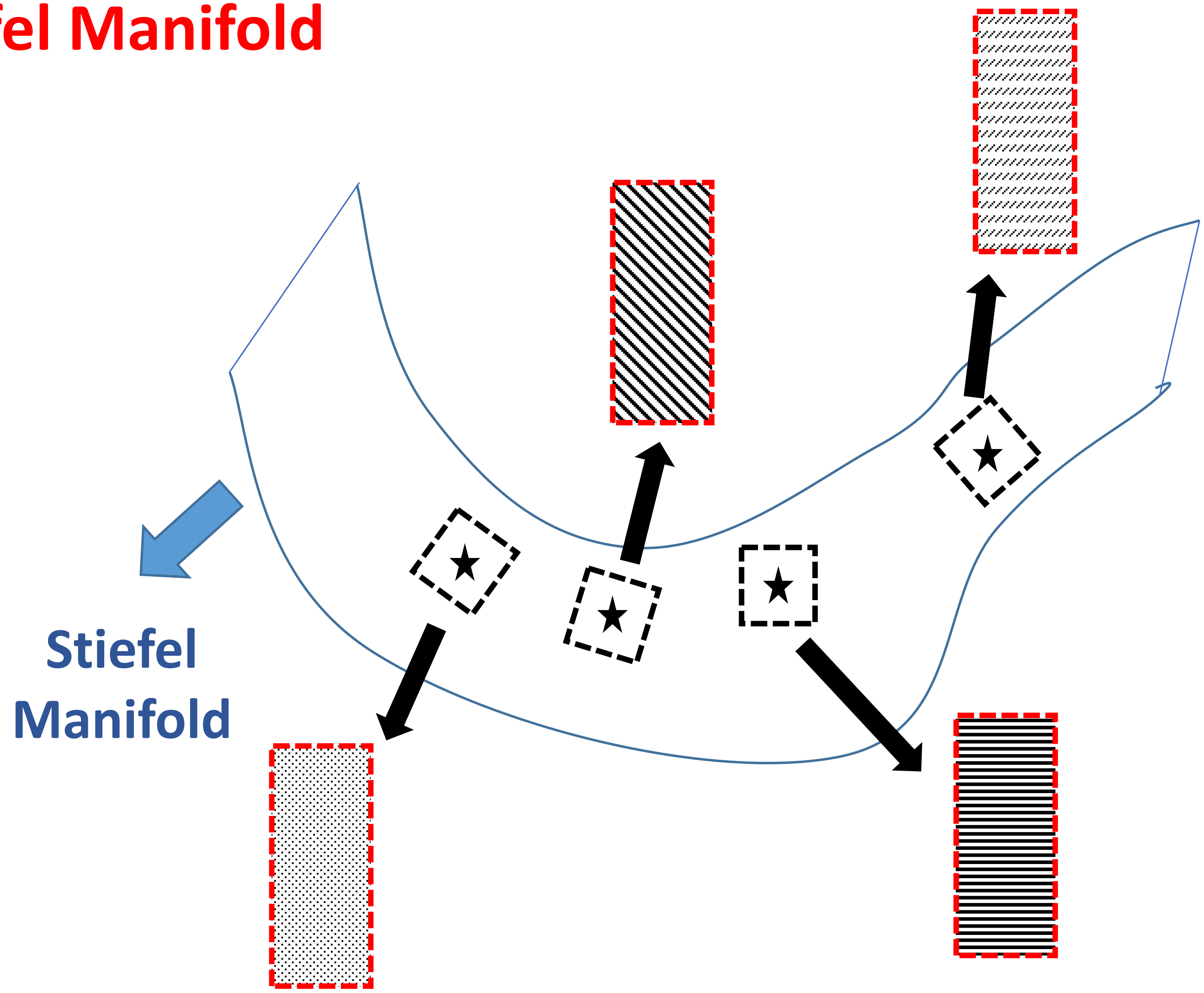
All possible $\boldsymbol{U_k}$



Stiefel manifold

Stiefel Manifold

❖ **Stiefel Manifold**

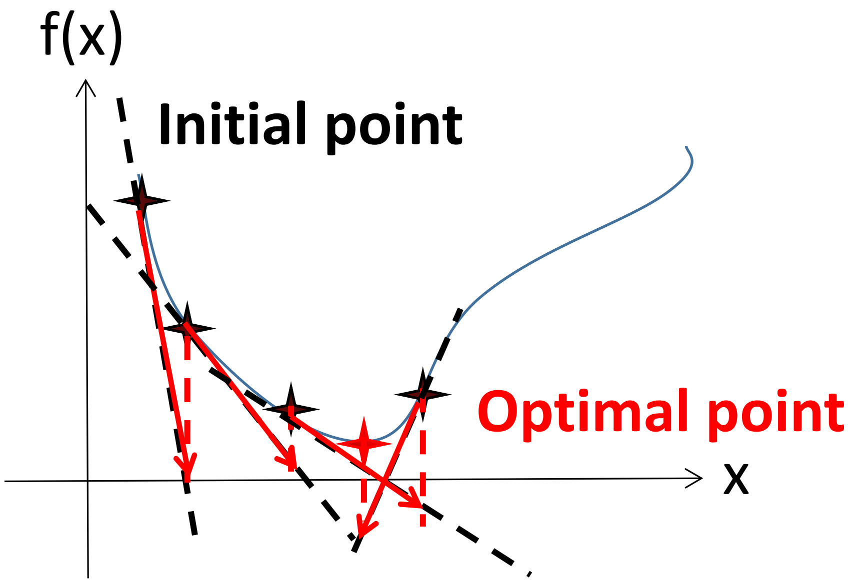


Stiefel-Manifold based Optimization

- **Finding the optimal U_k on Stiefel Manifold**

$$\min: f(U_k)$$

Recall: **Gradient Descent Method over real values**



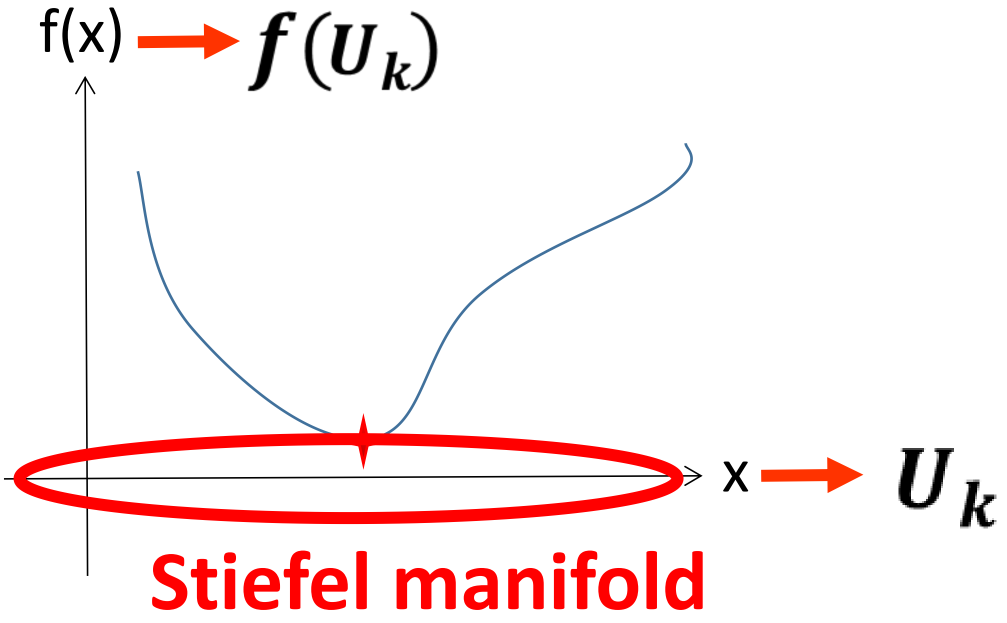
- **Can we make an analogy?**

Stiefel-Manifold based Optimization

- **Finding the optimal U_k on Stiefel Manifold**

$$\min: f(U_k)$$

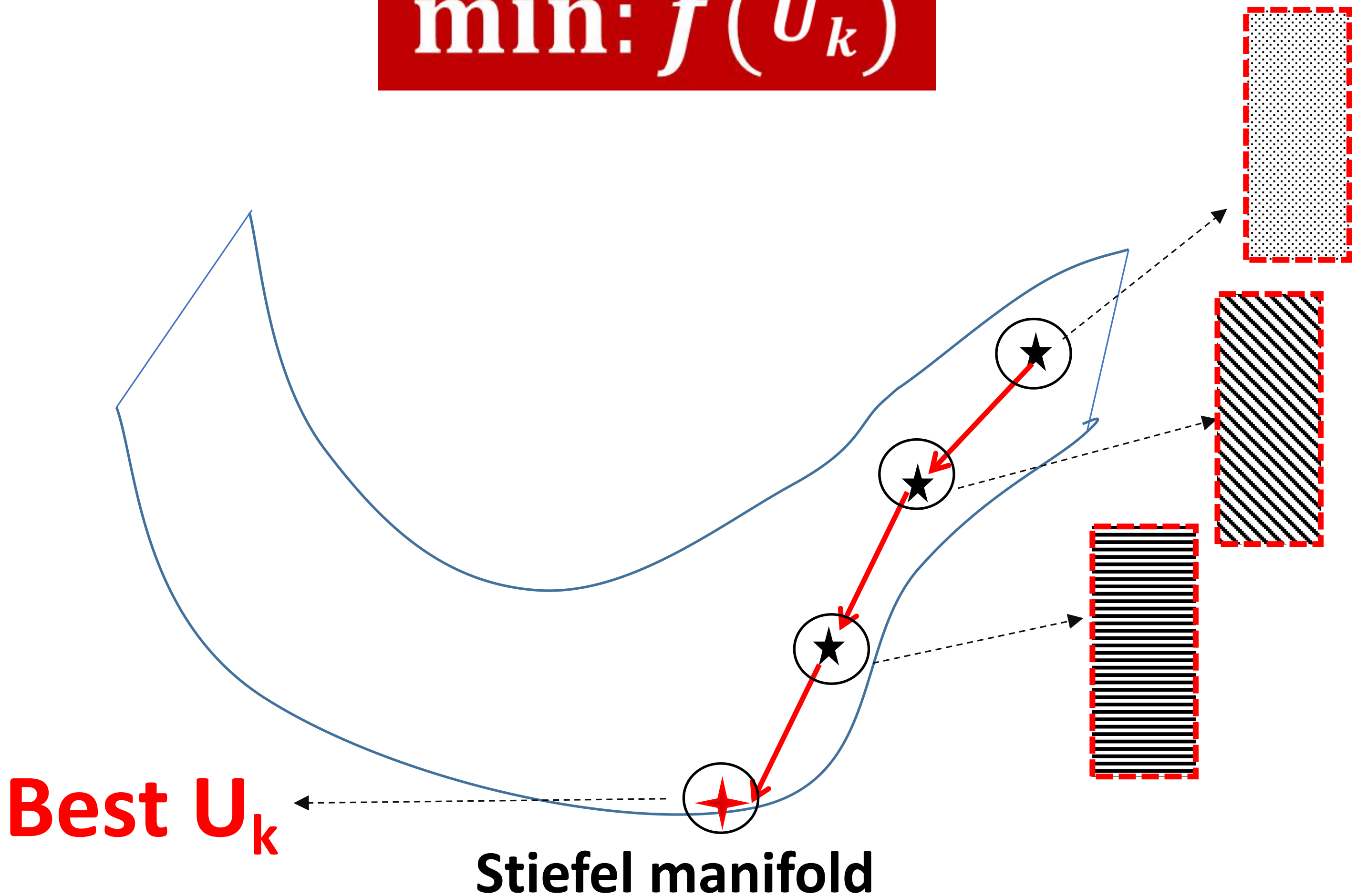
Stiefel-Manifold based Gradient Descent



Stiefel-Manifold based Optimization

- **Finding the optimal U_k**

$$\min: f(U_k)$$



Stiefel-Manifold based Optimization

Design Details

Stiefel-Manifold based Gradient Descent

$$\min_{\hat{A}} \| P_\Omega(A) - P_\Omega(\hat{A}) \| \Rightarrow \min_{\substack{U_k : m\times k \\ w_j : k\times 1}} \boxed{\sum_{j=1}^{n} \|[U_k w_j]_\Omega - [a_j]_\Omega\|_2^2}$$

$$F(U_k, w_1, \ldots, w_n)$$

Descent Direction $\nabla_{U_k} F$

Objective Simplification

$$\underbrace{\sum_{j=1}^{n} \|[U_k w_j]_\Omega - [a_j]_\Omega\|_2^2}_{F(U_k, w_1, \ldots, w_n)} \xrightarrow[\text{Convexity}]{} \underbrace{\min_{x_j} \|[U_k x_j]_\Omega - [a_j]_\Omega\|_2^2}_{F(U_k)}$$

Stiefel-Manifold based Optimization

- **Design Details**

Stiefel-Manifold based Gradient Descent

$$\min_{\hat{A}} \| P_\Omega(A) - P_\Omega(\hat{A}) \| \Rightarrow \min_{\substack{U_k : m \times k \\ w_j : k \times 1}} \sum_{j=1}^{n} ||[U_k w_j]_\Omega - [a_j]_\Omega||_2^2$$

- **Iteration Equation**

$$U_{t+1} = U_t + 2\eta_t \frac{r_t w_t^T}{||r_t|| \, ||w_t||}$$

$$r_t = P_\Omega(a_q - U_t w_t)$$

- **Step Size**

$$\eta_t = \frac{1}{2} \frac{||r_t||}{||w_t||}$$

Main Algorithm

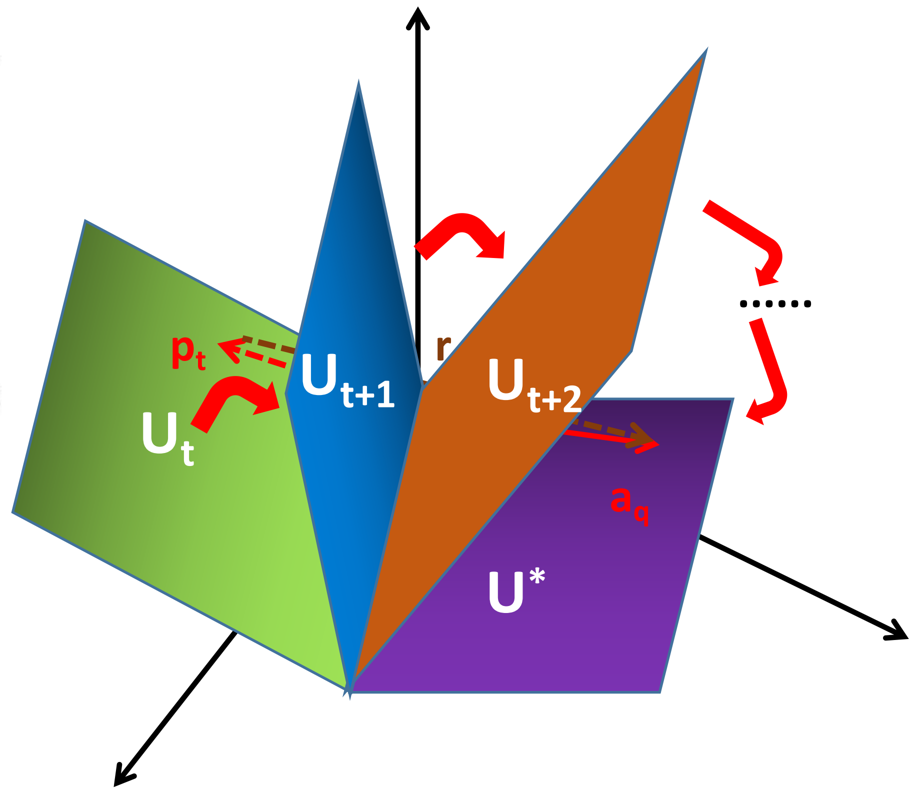
- **SSOA**: Streamlined Stiefel-manifold Optimization Algorithm

Algorithm 1: Streamlined Stiefel-manifold Optimization Algorithm (SSOA)

Input:
An initial column-orthonormal $m \times k$ matrix U_0;
sample set Ω, $m \times n$ sample matrix $P_\Omega(A)$;
maximum number of iteration T.

Output:
Estimated matrix $\hat{A}$.

$t = 0$;
while $t < T$ **do**
  Randomly choose a column index $q \in \{1, 2, ..., n\}$, get $[a_q]_\Omega$;
  $w_t = ([U_t]_\Omega^T [U_t]_\Omega)^{-1} [U_t]_\Omega [a_q]_\Omega$;
  $p_t = U_t w_t$;
  $r_t = P_\Omega(a_q - p_t)$;
  $U_{t+1} = U_t + \frac{r_t w_t^T}{||w_t||^2}$;
  $t = t + 1$;
end while
$U = U_t$;
for each $i \in \{1, 2, ..., n\}$ **do**
  $\hat{a}_i = U([U]_\Omega^T [U]_\Omega)^{-1} [U]_\Omega [a_i]_\Omega$;
end for
$\hat{A} = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_n]$.



Recover the fingerprints

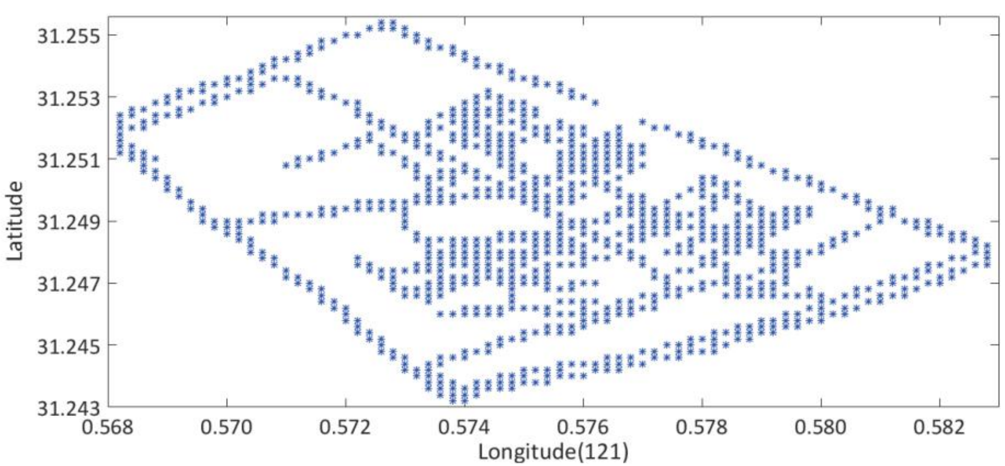
Outline

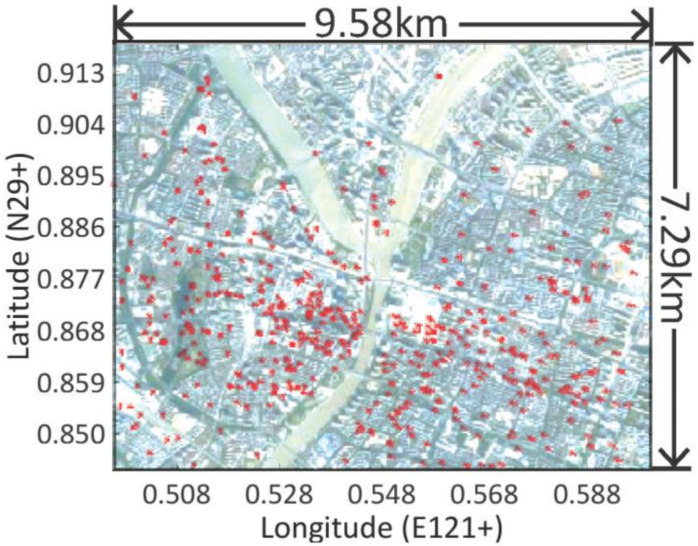
- Introduction & Motivation
- Problem Formulation
- Algorithm Design
- **Experimental Validation**
- Conclusion

Experimental Validation

❖ Datasets

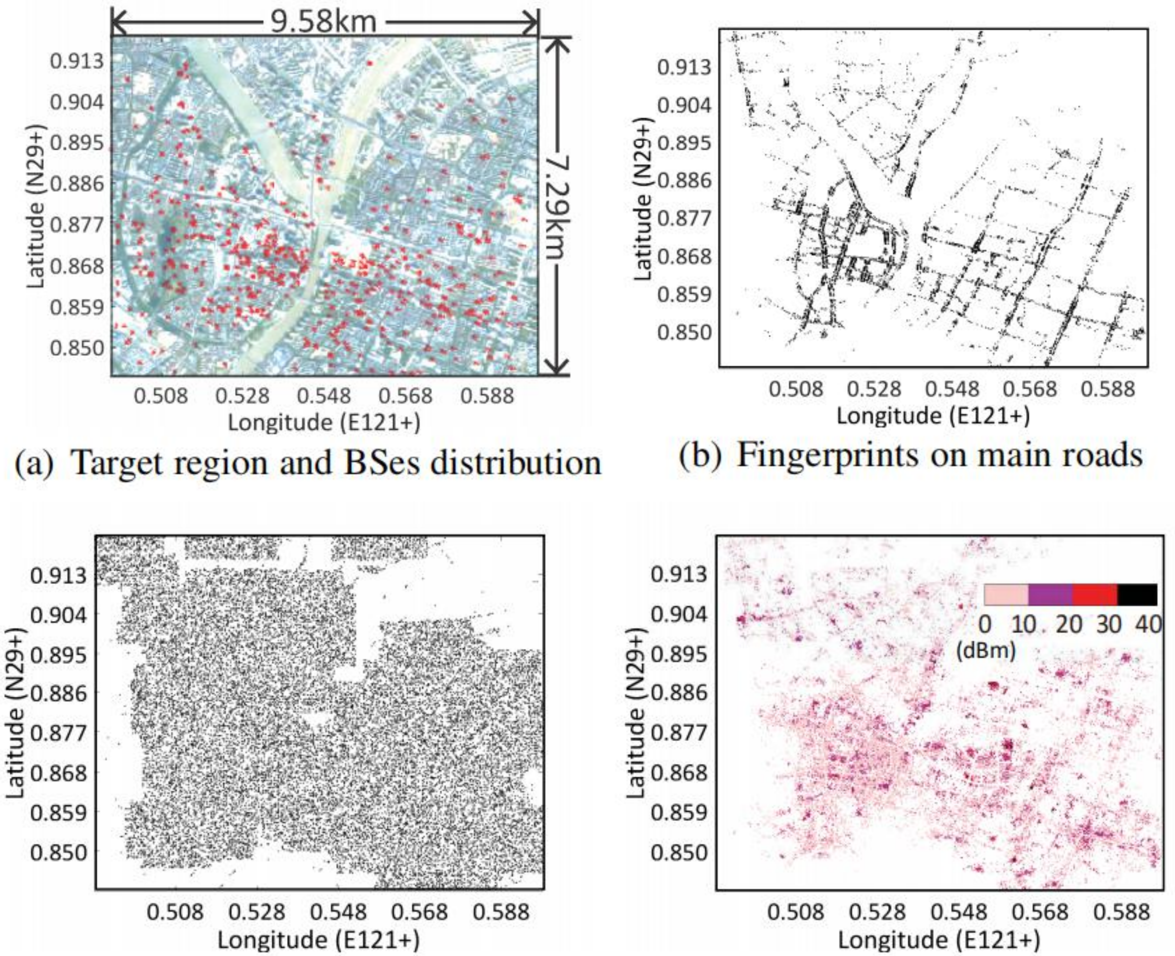
- **2.2 km², ~60,000 records, Shanghai, China.**
- **69.8 km2, ~8,820,000 records, Ningbo, China.**





Experimental Validation

❖ **Fingerprint Prediction**



(a) Target region and BSes distribution

(b) Fingerprints on main roads

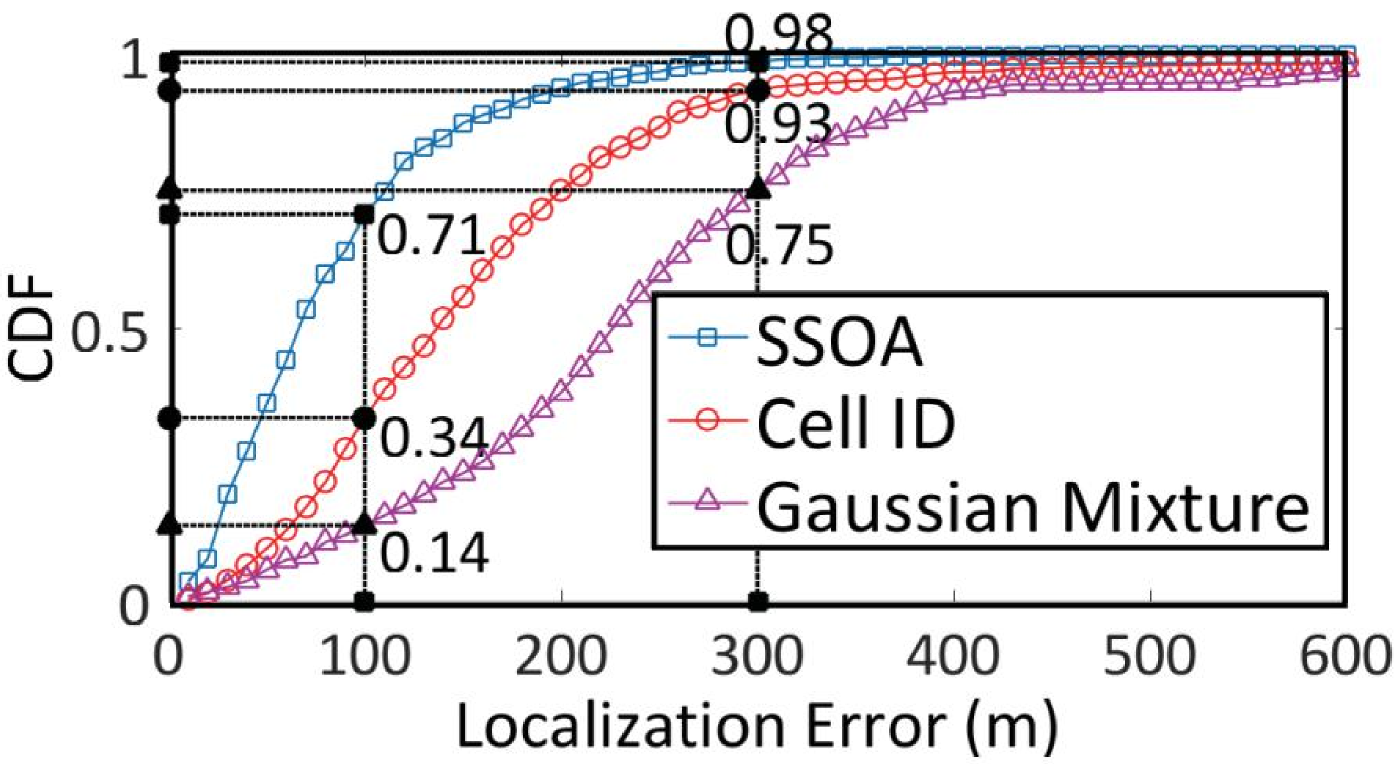
(c) Predicted fingerprints

(d) Prediction errors distribution

Experimental Validation

❖ Fingerprinting Localization

- Algorithms in comparison: **Cell ID, Gaussian Mixture Model.**



Experimental Validation

THE WIRELESS E911 LOCATION ACCURACY REQUIREMENTS [2]

For terminal-based and terminal-assisted positioning:

- 50m, 67% – within 50m for 67% of all calls measured at country level
- 150m, 95% – within 150m for 95% of all calls measured at county level

For network-based positioning:

- 100m, 67% – within 100m for 67% of all calls measured at county level
- 300m, 90% – within 300m for 90% of all calls measured at county level

Carriers must provide location, together with confidence and uncertainty data, for all emergency calls at the PSAPs.

<100m 67%

<300m 90%

	<100m	<300m
SSOA	71%	98%
Cell ID	34%	93%
GMM	14%	75%

[1] Ericsson White Paper, Postioning with LTE, Online: http://www.sharetechnote.com/Docs/WP-LTE-positioning.pdf

Outline

- Introduction & Motivation
- Problem Formulation
- Algorithm Design
- Experimental Validation
- **Conclusion**

Conclusion

- Proposed a **Streamlined Stiefel-manifold Optimization Algorithm (SSOA)** based on Gradient Descent.
- Validated our proposed mechanisms based on **large-scale real datasets**.

Please refer to the paper for more details about:

1. **Convergence** analysis of SSOA.
2. **Sliding window** mechanism for applying SSOA in real cases.
3. Theoretical analysis of **determining k** in U_k.
4. Experiment of fingerprint prediction on the **smaller dataset**.

Thanks for Listening!

Q&A