DATA RECOVERY ON ENCRYPTED DATABASES WITH k-NEAREST NEIGHBOR QUERY LEAKAGE

EVGENIOS M. KORNAROPOULOS CHARALAMPOS PAPAMANTHOU ROBERTO TAMASSIA

Full version: https://eprint.iacr.org/2018/719
WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

18.2.8. KNearestNeighborProcess

The KNearestNeighborProcess performs a K Nearest Neighbor search on a Geomesa feature collection using another feature collection as input. Return k neighbors for each point in the input data set. If a point is the nearest neighbor of multiple points of the input data set, it is returned only once.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputFeatures</td>
<td>Input feature collection that defines the KNN search.</td>
</tr>
<tr>
<td>dataFeatures</td>
<td>The data set to query for matching features.</td>
</tr>
<tr>
<td>numDesired</td>
<td>K: number of nearest neighbors to return.</td>
</tr>
<tr>
<td>estimatedDistance</td>
<td>Estimate of Search Distance in meters for K neighbors—used to set the granularity of the search.</td>
</tr>
<tr>
<td>maxSearchDistance</td>
<td>Maximum search distance in meters—used to prevent runaway queries of the entire table.</td>
</tr>
</tbody>
</table>

18.2.8.1. K-Nearest-Neighbor example (XML)

KNNProcess.wps.xml is a geoserver WPS call to the GeoMesa KNearestNeighborProcess. It is here chained with a Query process (see Chaining Processes) in order to avoid points related to the same Id to be matched by the request. It can be run with the following curl call:

```
curl -v -u admin:geoserver -H "Content-Type: text/xml" -d@KNNProcess_wps.xml localhost:8080/geoserver/wps
```
INTRO

WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

OBJECT-RELATIONAL DBMS

27.2. Index-based KNN

“KNN” stands for “K nearest neighbours”, where “K” is the number of neighbours you are looking for.

KNN is a pure index based nearest neighbour search. By walking up and down the index, the search can find the nearest candidate geometries without using any magical search radius numbers, so the technique is suitable and high performance even for very large tables with highly variable data densities.

Note

The KNN feature is only available on PostGIS 2.0 with PostgreSQL 9.1 or greater.

The KNN system works by evaluating distances between bounding boxes inside the PostGIS R-Tree index.

Because the index is built using the bounding boxes of geometries, the distances between any geometries that are not points will be inexact: they will be the distances between the bounding boxes of geometries.

The syntax of the index-based KNN query places a special “index-based distance operator” in the ORDER BY clause of the query, in this case “<>“. There are two index-based distance operators,

- <> means “distance between box centers”
- # means “distance between box edges”

One side of the index-based distance operator must be a literal geometry value. We can get away with a subquery that returns as single geometry, or we could include a WKT geometry instead.

```sql
-- Closest 10 streets to Broad Street station are 7
SELECT streets.gid, streets.name
FROM
```
INTRO

WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

OBJECT-RELATIONAL DBMS

CLOUD SERVICES

**Nearest neighbor search**

IBM Cloudant Geo supports Nearest Neighbor search, which is known as NN search. If provided, the `nearest=true` search returns all results by sorting their distances to the center of the query geometry. This geometric relation `nearest=true` can be used either with all the geometric relations described earlier, or alone.

For example, a police officer might search five crimes that occurred near a specific location by typing the query in the following example.

Example query to find nearest five crimes against a specific location:

```
https://education.cloudant.com/crimes/_design/geoidd/_geo/geoiddx?q=POINT(-71.053712 42.348195)
```

**Tip:** The `nearest=true` search can change the semantics of an IBM Cloudant Geo search. For example, without `nearest=true` in the example query, the results include only GeoJSON documents that have coordinates equal to the query point (-71.053712 42.348195) or an empty results set. However, by using the `nearest=true` search, the results include all GeoJSON documents in the database whose order is measured by the distance to the query point.

One side of the index-based distance operator must be a literal geometry value. We can get away with a subquery that returns as single geometry, or we could include a WKT geometry instead.

```sql
-- Closest 10 streets to Broad Street station are 7
SELECT streets.gid, streets.name
FROM
```
SETUP

k-NEAREST NEIGHBORS

Records:

$s_0$  $s_1$  $s_2$  $s_3$  $s_4$  $s_5$

$\alpha$  $v_0$  $v_1$  $v_2$  $v_3$  $v_4$  $v_5$  $\beta$
SETUP

**k-NEAREST NEIGHBORS**

Records: $s_0, s_1, s_2, s_3, s_4, s_5$

Diagram showing a line with points $v_0, v_1, v_2, v_3, v_4, v_5$ and a query point $q$. The records are colored red, white, blue, and green, indicating different classes or categories.
k-NEAREST NEIGHBORS

Records:

- \( s_0 \)
- \( s_1 \)
- \( s_2 \)
- \( s_3 \)
- \( s_4 \)
- \( s_5 \)

\( \{s_1, s_2, s_3\} \)
SETUP
VORONOI DIAGRAMS

\[ \{s_1, s_2, s_3\} \]
SETUP
VORONOI DIAGRAMS

$\alpha$

$\beta$

$\{s_0, s_1, s_2\}$

$\{s_1, s_2, s_3\}$

$\{s_2, s_3, s_4\}$

$\{s_3, s_4, s_5\}$

$v_0, v_1, v_2, v_3, v_4, v_5$

$b_{0,3}$

$b_{1,4}$

$b_{2,5}$
Voronoi Diagrams

Voronoi Segments

Voronoi Edges

\[ b_{0,3} \quad b_{1,4} \quad b_{2,5} \]

Response

\[ \{s_0, s_1, s_2\} \quad \{s_1, s_2, s_3\} \quad \{s_2, s_3, s_4\} \quad \{s_3, s_4, s_5\} \]
SETUP
ENCRYPTED SEARCH

Client

Server
setup

encrypted search

Client

Server

PRF

\[ PRF_K(t) = t \]

tokens

responses
SETUP
ENCRYPTED SEARCH

Client

Server

Tokens

PRF_\(K(\bullet) = t\)

PRF_\(K(\bullet) = t'\)

PRF_\(K(\bullet) = t''\)

PRF_\(K(\bullet) = t\)

Responses

PRF_\(K(\bullet) = t\)

PRF_\(K(\bullet) = t'\)

PRF_\(K(\bullet) = t''\)

PRF_\(K(\bullet) = t\)
SETUP

ENCRYPTED SEARCH

Client

Server

Tokens

PRF\_K(\textcircled{1}) = t

PRF\_K(\textcircled{2}) = t'

PRF\_K(\textcircled{3}) = t''

PRF\_K(\textcircled{4}) = t

Responses
ENCRYPTED SEARCH

Client

Server

Tokens

\[ PRF_K(t) = t \]

\[ PRF_K(t') = t' \]

\[ PRF_K(t'') = t'' \]

\[ PRF_K(t) = t \]

Search Pattern Leakage

Responses
ENCRYPTED SEARCH

**SETUP**

Client

Server

**Tokens**

\[ \text{PRF}_K(\_\_\_\_\_\_\_, ) = t \]

\[ \text{PRF}_K(\_, _\_\_\_\_\_, ) = t' \]

\[ \text{PRF}_K(\_, _\_, _\_\_\_\_, ) = t'' \]

\[ \text{PRF}_K(\_, _\_, _\_, _\_\_\_, ) = t \]

**Responses**

**Search Pattern Leakage**

**Access Pattern Leakage**
OUR CONTRIBUTIONS

OVERVIEW

k-NN EXACT RECONSTRUCTION
k-NN EXACT RECONSTRUCTION

**ORDERED RESPONSES:** Possible when all encrypted queries are issued

**UNORDERED RESPONSES:** Impossible due to many reconstructions
OUR CONTRIBUTIONS

OVERVIEW

**k-NN EXACT RECONSTRUCTION**

**ORDERED RESPONSES:** Possible when all encrypted queries are issued

**UNORDERED RESPONSES:** Impossible due to many reconstructions

**k-NN APPROXIMATE RECONSTRUCTION**
OUR CONTRIBUTIONS
OVERVIEW

**k-NN EXACT RECONSTRUCTION**

**ORDERED RESPONSES:** Possible when all encrypted queries are issued

**UNORDERED RESPONSES:** Impossible due to many reconstructions

**k-NN APPROXIMATE RECONSTRUCTION**

**ORDERED RESPONSES:** Approximate reconstruction when not all encrypted queries are issued

**UNORDERED RESPONSES:** Even with many reconstructions approximate with bounded error
OVERVIEW

k-NN EXACT RECONSTRUCTION

ORDERED RESPONSES: Possible when all encrypted queries are issued

UNORDERED RESPONSES: Impossible due to many reconstructions

k-NN APPROXIMATE RECONSTRUCTION

ORDERED RESPONSES: Approximate reconstruction when not all encrypted queries are issued

UNORDERED RESPONSES: Even with many reconstructions approximate with bounded error
Assumptions of the Attack

Boundaries:
- Known boundaries $\alpha$ and $\beta$

Static:
- No updates in the database

Uniformity:
- Queries are generated uniformly at random from $[\alpha, \beta]$
UNORDERED RESPONSES
EXACT RECONSTRUCTION

Best Case Scenario for the Adversary

\[ k=2 \]
\[ \{s_0, s_1\} \quad b_{0,2} \quad \{s_1, s_2\} \quad b_{1,3} \quad \{s_2, s_3\} \quad b_{2,4} \quad \{s_3, s_4\} \quad b_{3,5} \quad \{s_4, s_5\} \]
Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

\[ k=2 \quad \{s_0, s_1\} \quad b_{0,2} \quad \{s_1, s_2\} \quad b_{1,3} \quad \{s_2, s_3\} \quad b_{2,4} \quad \{s_3, s_4\} \quad b_{3,5} \quad \{s_4, s_5\} \]
Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

Valid Reconstruction $DB_1$
UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Valid Reconstruction
DB₁

Valid Reconstruction
DB₂

Best Case Scenario for the Adversary

k=2

{\(s₀, s₁\)}

{\(s₁, s₂\)}

{\(s₂, s₃\)}

{\(s₃, s₄\)}

{\(s₄, s₅\)}
Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

Valid Reconstruction $DB_1$

Valid Reconstruction $DB_2$
Impossible to achieve Exact Reconstruction

Valid Reconstruction $DB_1$

Best Case Scenario for the Adversary

Valid Reconstruction $DB_2$
Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

Valid Reconstruction DB₁

Valid Reconstruction DB₂
UNORDERED RESPONSES

EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

Valid Reconstruction DB

Valid Reconstruction DB

$\{s_0, s_1\}$ $\{s_1, s_2\}$ $\{s_2, s_3\}$ $\{s_3, s_4\}$ $\{s_4, s_5\}$

$\alpha$ $\beta$
UNORDERED RESPONSES

EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario
for the Adversary

Valid Reconstruction
DB₁

Valid Reconstruction
DB₂

$k=2$

{$s₀, s₁$}        {$s₁, s₂$}        {$s₂, s₃$}        {$s₃, s₄$}        {$s₄, s₅$}

$α$         $b_{0,2}$         $b_{1,3}$         $b_{2,4}$         $b_{3,5}$         $β$

$β$
Many reconstructions that explain the Voronoi Diagram
Since there are **MANY** reconstructions and the exact recovery is **IMPOSSIBLE**, the encrypted values must be safe...
Since there are **MANY** reconstructions and the exact recovery is **IMPOSSIBLE**, the encrypted values must be safe...

**Answer:** We can still compute an reconstruction that is **VERY CLOSE** to the encrypted DB.
In case all queries are issued:

The length of each Voronoi segments

\[ b_{0,2} \quad b_{1,3} \quad b_{2,4} \quad b_{3,5} \]

\[ \{s_0, s_1\} \quad \{s_1, s_2\} \quad \{s_2, s_3\} \quad \{s_3, s_4\} \quad \{s_4, s_5\} \]
In case all queries are issued:

The length of each Voronoi segments

Uniform Query Distribution: Estimate via Concentration Bounds on Multinomials
In case all queries are issued:

The length of each Voronoi segments

Goal:

Characterize the set of all valid reconstructions that explain the Voronoi Diagram
In case all queries are issued:

The length of each Voronoi segments

Goal:

Characterize the set of all valid reconstructions that explain the Voronoi Diagram

What’s Next:

Intuitive characterization = rigorous reconstruction guarantees
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:
UNORDERED RESPONSES

APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

*Use geometry of bisectors to define unknowns*
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
\[ v_4 = 2b_{2,4} - v_2 \]
MODELING ALL RECONSTRUCTIONS:

*Use geometry of bisectors to define unknowns*

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
\[ v_4 = 2b_{2,4} - v_2 \]
Modeling All Reconstructions: Use geometry of bisectors to define unknowns

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
\[ v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0 \]
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
\[ v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0 \]
\[ v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0 \]
\[ v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0 \]

Half of the \( U_q \) as a function of unknown \( \xi_0 \)
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns

\[ v_0 = b_{0,2} - \xi_0 \]
\[ v_2 = b_{0,2} + \xi_0 \]
\[ v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0 \]
\[ v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0 \]
\[ v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0 \]

Half of the \( U_i \) as a function of unknown \( \xi_0 \)

\[ v_1 = b_{1,3} - \xi_1 \]
\[ v_3 = b_{1,3} + \xi_1 \]
\[ v_5 = 2b_{3,5} - v_3 = 2b_{3,5} - b_{1,3} - \xi_1 \]
\[ v_7 = 2b_{5,7} - v_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1 \]
\[ v_9 = 2b_{7,9} - v_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1 \]

Other half of the \( U_i \) as a function of unknown \( \xi_1 \)
Modeling All Reconstructions:

**Use geometry of bisectors to define unknowns**

Reduced the space of reconstructions from n-dimensions to 2-dimensions

\[ v_0 = b_{0.2} - \xi_0 \]
\[ v_2 = b_{0.2} + \xi_0 \]
\[ v_4 = 2b_{2.4} - v_2 = 2b_{2.4} - b_{0.2} - \xi_0 \]
\[ v_6 = 2b_{4.6} - v_4 = 2b_{4.6} - 2b_{2.4} + b_{0.2} + \xi_0 \]
\[ v_8 = 2b_{6.8} - v_6 = 2b_{6.8} - 2b_{4.6} + 2b_{2.4} - b_{0.2} - \xi_0 \]

**Half of the** \( U_i \) **as a function of unknown** \( \xi_0 \)

\[ v_1 = b_{1.3} - \xi_1 \]
\[ v_3 = b_{1.3} + \xi_1 \]
\[ v_5 = 2b_{3.5} - v_3 = 2b_{3.5} - b_{1.3} - \xi_1 \]
\[ v_7 = 2b_{5.7} - v_5 = 2b_{5.7} - 2b_{3.5} + b_{1.3} + \xi_1 \]
\[ v_9 = 2b_{7.9} - v_7 = 2b_{7.9} - 2b_{5.7} + 2b_{3.5} - b_{1.3} - \xi_1 \]

**Other half of the** \( U_i \) **as a function of unknown** \( \xi_1 \)
Modeling All Reconstructions:

Ordering Constraints:

\[ v_0 < v_1 \]
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

\[ v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1} \text{, where } c_{0,1} = (b_{1,3} - b_{0,2}) \]
MODELING ALL RECONSTRUCTIONS:

ORDERING CONSTRAINTS:

\[ v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1}, \text{ where } c_{0,1} = (b_{1,3} - b_{0,2}) \]
\[ v_1 < v_2 \Rightarrow -\xi_0 - \xi_1 < c_{1,2}, \text{ where } c_{1,2} = -(b_{1,3} - b_{0,2}) \]
\[ v_2 < v_3 \Rightarrow \xi_0 - \xi_1 < c_{2,3}, \text{ where } c_{2,3} = (b_{1,3} - b_{0,2}) \]
\[ v_3 < v_4 \Rightarrow \xi_0 + \xi_1 < c_{3,4}, \text{ where } c_{3,4} = (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2}) \]
\[ v_4 < v_5 \Rightarrow -\xi_0 + \xi_1 < c_{4,5}, \text{ where } c_{4,5} = 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2}) \]
\[ v_5 < v_6 \Rightarrow -\xi_0 - \xi_1 < c_{5,6}, \text{ where } c_{5,6} = 2(b_{4,6} - b_{3,5}) - (b_{2,4} - b_{0,2}) - (b_{2,4} - b_{1,3}) \]
\[ v_6 < v_7 \Rightarrow \xi_0 - \xi_1 < c_{6,7}, \text{ where } c_{6,7} = 2(b_{5,7} - b_{4,6}) - 2(b_{3,5} - b_{2,4}) + (b_{1,3} - b_{0,2}) \]
\[ v_7 < v_8 \Rightarrow \xi_0 + \xi_1 < c_{7,8}, \text{ where } c_{7,8} = 2(b_{6,8} - b_{5,7}) - 2(b_{4,6} - b_{3,5}) + (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2}) \]
\[ v_8 < v_9 \Rightarrow -\xi_0 + \xi_1 < c_{8,9}, \text{ where } c_{8,9} = 2(b_{7,9} - b_{6,8}) - 2(b_{5,7} - b_{4,6}) + 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2}) \]
Modeling All Reconstructions:

**Ordering Constraints:**
\[
\begin{align*}
  v_0 < v_1 & \Rightarrow -\xi_0 + \xi_1 < c_{0,1}, \text{ where } c_{0,1} = (b_{1.3} - b_{0.2}) \\
  v_1 < v_2 & \Rightarrow -\xi_0 - \xi_1 < c_{1.2}, \text{ where } c_{1.2} = -(b_{1.3} - b_{0.2}) \\
  v_2 < v_3 & \Rightarrow \xi_0 - \xi_1 < c_{2.3}, \text{ where } c_{2.3} = (b_{1.3} - b_{0.2}) \\
  v_3 < v_4 & \Rightarrow \xi_0 + \xi_1 < c_{3.4}, \text{ where } c_{3.4} = (b_{2.4} - b_{1.3}) + (b_{2.4} - b_{0.2}) \\
  v_4 < v_5 & \Rightarrow -\xi_0 + \xi_1 < c_{4.5}, \text{ where } c_{4.5} = 2(b_{3.5} - b_{2.4}) - (b_{1.3} - b_{0.2}) \\
  v_5 < v_6 & \Rightarrow -\xi_0 - \xi_1 < c_{5.6}, \text{ where } c_{5.6} = 2(b_{4.6} - b_{3.5}) - (b_{2.4} - b_{0.2}) - (b_{2.4} - b_{1.3}) \\
  v_6 < v_7 & \Rightarrow \xi_0 - \xi_1 < c_{6.7}, \text{ where } c_{6.7} = 2(b_{5.7} - b_{4.6}) - 2(b_{3.5} - b_{2.4}) + (b_{1.3} - b_{0.2}) \\
  v_7 < v_8 & \Rightarrow \xi_0 + \xi_1 < c_{7.8}, \text{ where } c_{7.8} = 2(b_{6.8} - b_{5.7}) - 2(b_{4.6} - b_{3.5}) + (b_{2.4} - b_{1.3}) + (b_{2.4} - b_{0.2}) \\
  v_8 < v_9 & \Rightarrow -\xi_0 + \xi_1 < c_{8.9}, \text{ where } c_{8.9} = 2(b_{7.9} - b_{6.8}) - 2(b_{5.7} - b_{4.6}) + 2(b_{3.5} - b_{2.4}) - (b_{1.3} - b_{0.2}) \\
\end{align*}
\]

**Boundary Constraints:**
\[
\begin{align*}
  \alpha < v_0 & \Rightarrow \xi_0 < c_{\alpha,0}, \text{ where } c_{\alpha,0} = b_{0.2} - \alpha \\
  v_9 < \beta & \Rightarrow \xi_1 > c_{9,\beta}, \text{ where } c_{9,\beta} = 2b_{7.9} - 2b_{5.7} + 2b_{3.5} - b_{1.3} - \beta \\
\end{align*}
\]
“Squeezed” the seemingly large space of valid reconstructions into a small polygon.
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: \( v' = (v'_0, \ldots, v'_{n-1}) \)
Reconstr. DB: \( v'' = (v''_0, \ldots, v''_{n-1}) \)
Reconstruction Error between $v', v''$

$$\max_{i \in [0,n-1]} |v'_i - v''_i| \leq \text{diam}(F_v)$$

Original DB: $v = (v'_0, \ldots, v'_{n-1})$

Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$

Maximum Error
Reconstruction Error between $v', v''$

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \leq \text{diam}(F_v)$$
**Reconstruction Error between** $v', v''$

\[
\max_{i \in [0,n-1]} |v_i' - v_i''| \leq \text{diam}(F_v)
\]
Reconstruction Error between $v', v''$

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \leq diam(F_v)$$

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$
Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$

Our Reconstruction

The worst case reconstruction between $v''$ and every DB in $F_v$ is upper-bounded by $\frac{diam(F_v)}{2}$.
Case $k=3$

$k$-NN queries $\rightarrow F_v$ is a polytope in $k$-dimensional space
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION

Case $k=3$

$k$-NN queries $\rightarrow F_v$ is a polytope in $k$-dimensional space
EVALUATION
ORDERED & UNORDERED RESPONSES
1-31 October 2009

- Geolocation of politician Spitz
- Simulated k-NN Leakage from queries on his location DB
EVALUATION
ORDERED & UNORDERED RESPONSES

1-31 October 2009

- Geolocation of politician Spitz
- Simulated k-NN Leakage from queries on his location DB
1-31 October 2009

- Geolocation of politician Spitz
- Simulated k-NN Leakage from queries on his location DB

<table>
<thead>
<tr>
<th>1-31 October, ( m = 250 \cdot 10^6, n = 183 )</th>
<th>diameter</th>
<th>Absolute Error</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 2 )</td>
<td>1.8</td>
<td>1.0</td>
<td>70%</td>
</tr>
<tr>
<td>( k = 5 )</td>
<td>6.4</td>
<td>1.4</td>
<td>95%</td>
</tr>
<tr>
<td>( k = 8 )</td>
<td>12.8</td>
<td>1.4</td>
<td>95%</td>
</tr>
</tbody>
</table>
OUR CONTRIBUTIONS

CONCLUSIONS

k-NN EXACT RECONSTRUCTION

ORDERED RESPONSES: Possible when all encrypted queries are issued

UNORDERED RESPONSES: Impossible due to many reconstructions

k-NN APPROXIMATE RECONSTRUCTION

ORDERED RESPONSES: Approximate reconstruction when not all encrypted queries are issued

UNORDERED RESPONSES: Even with many reconstructions approximate with bounded error

Thank you!