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
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
SELECT 
    streets.gid, 
    streets.name 
FROM 
```
## INTRO

**WHO CARES ABOUT k-NN?**

#### COLUMN-ORIENTED DBMS

#### OBJECT-RELATIONAL DBMS

#### CLOUD SERVICES

<table>
<thead>
<tr>
<th>IBM Cloud</th>
<th>Catalog</th>
<th>Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Cloudant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEARN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Getting started tutorial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM Cloud Public Pricing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security and Compliance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other offerings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOW TO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tutorials</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovery and backup</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**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:

```sql
https://education.cloudant.com/crimes_design/geojson GEO/geoidx7q=POINT(71.053712)
```

### 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 (`71.0537124, 42.3481955`) or empty result sets. 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?
SELECT
    streets gid, streets.name
FROM
    streets
WHERE
    ST_Distance(streets.geom, 'POLYGON((-71.0537124 42.3481955))') < 0.0001;
```
Setup

k-Nearest Neighbors

Records:

\[ s_0 \quad s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \]

\[ v_0 \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \]
SETUP
k-NEAREST NEIGHBORS

Records:

s_0  s_1  s_2  s_3  s_4  s_5

\[ \alpha \quad v_0 \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad \beta \]
SETUP
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

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

\( \alpha \quad v_0 \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad \beta \)

\( \{s_0, s_1, s_2\} \quad \{s_1, s_2, s_3\} \quad \{s_2, s_3, s_4\} \quad \{s_3, s_4, s_5\} \)

\( b_{0,3} \quad b_{1,4} \quad b_{2,5} \)
SETUP
VORONOI DIAGRAMS

Voronoi Diagram

Voronoi Edges $b_{0,3}$

$\alpha$

$\nu_0 \nu_1 \nu_2$

Voronoi Segment

$\beta$

$\nu_3 \nu_4 \nu_5$

Response

$\{s_0, s_1, s_2\}$

$\{s_1, s_2, s_3\}$

$\{s_2, s_3, s_4\}$

$\{s_3, s_4, s_5\}$
SETUP

CLIENT

PRF\( K(\cdot) = t \)

PRF\( K(\cdot) = t' \)

PRF\( K(\cdot) = t'' \)

PRF\( K(\cdot) = t \)

SERVER

Tokens

Responses
Setup

Encrypted Search

Client

Server

Prf

Keys

Tokens

PRF_K(●) = t

PRF_K(●) = t'

PRF_K(●) = t''

PRF_K(●) = t

Responses

Search Pattern Leakage

Access Pattern Leakage
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
**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
UNORDERED RESPONSES

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

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

\[ k=2 \]
\[ \{s_0, s_1\} \]
\[ \{s_1, s_2\} \]
\[ \{s_2, s_3\} \]
\[ \{s_3, s_4\} \]
\[ \{s_4, s_5\} \]
Impossible to achieve Exact Reconstruction

Valid Reconstruction DB₁

Best Case Scenario for the Adversary
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\} $
UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

$k = 2$

Validation Response

Valid Reconstruction $DB_1$

Valid Reconstruction $DB_2$
Impossible to achieve Exact Reconstruction

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

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:
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

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

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

Use geometry of bisectors to define unknowns

Half of the $U_i$ as a function of unknown $\xi_0$

$\nu_0 = b_{0,2} - \xi_0$

$\nu_2 = b_{0,2} + \xi_0$

$\nu_4 = 2b_{2,4} - \nu_2 = 2b_{2,4} - b_{0,2} - \xi_0$

$\nu_6 = 2b_{4,6} - \nu_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0$

$\nu_8 = 2b_{6,8} - \nu_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0$

Other half of the $U_i$ as a function of unknown $\xi_1$

$\nu_1 = b_{1,3} - \xi_1$

$\nu_3 = b_{1,3} + \xi_1$

$\nu_5 = 2b_{3,5} - \nu_3 = 2b_{3,5} - b_{1,3} - \xi_1$

$\nu_7 = 2b_{5,7} - \nu_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1$

$\nu_9 = 2b_{7,9} - \nu_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1$
Modeling All Reconstructions:  

Use geometry of bisectors to define unknowns

Half of the $U_i$ as a function of unknown $\xi_0$

Other half of the $U_i$ as a function of unknown $\xi_1$

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

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

$v_1 = b_{1,3} - \xi_1$
$v_3 = b_{1,3} + \xi_1$
$v_5 = 2b_{3,5} - v_3 = 2b_{3,5} - b_{1,3} - \xi_1$
$v_7 = 2b_{5,7} - v_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1$
$v_9 = 2b_{7,9} - v_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1$
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

\[ v_0 < v_1 \]
Modeling All Reconstructions:

Ordering Constraints:

\[ v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1} \], where \( c_{0,1} = (b_{1,3} - b_{0,2}) \)
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

\[ v_0 < v_1 \implies -\xi_0 + \xi_1 < c_{0,1}, \text{ where } c_{0,1} = (b_{1,3} - b_{0,2}) \]
\[ v_1 < v_2 \implies -\xi_0 - \xi_1 < c_{1,2}, \text{ where } c_{1,2} = -(b_{1,3} - b_{0,2}) \]
\[ v_2 < v_3 \implies \xi_0 - \xi_1 < c_{2,3}, \text{ where } c_{2,3} = (b_{1,3} - b_{0,2}) \]
\[ v_3 < v_4 \implies \xi_0 + \xi_1 < c_{3,4}, \text{ where } c_{3,4} = (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2}) \]
\[ v_4 < v_5 \implies -\xi_0 + \xi_1 < c_{4,5}, \text{ where } c_{4,5} = 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2}) \]
\[ v_5 < v_6 \implies -\xi_0 - \xi_1 < c_{5,6}, \text{ where } c_{5,6} = 2(b_{4,6} - b_{3,5}) - (b_{2,4} - b_{0,2}) - (b_{2,4} - b_{1,3}) \]
\[ v_6 < v_7 \implies \xi_0 - \xi_1 < c_{6,7}, \text{ where } c_{6,7} = 2(b_{5,7} - b_{4,6}) - 2(b_{3,5} - b_{2,4}) + (b_{1,3} - b_{0,2}) \]
\[ v_7 < v_8 \implies \xi_0 + \xi_1 < c_{7,8}, \text{ where } c_{7,8} = 2(b_{6,8} - b_{5,7}) - 2(b_{4,6} - b_{3,5}) + (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2}) \]
\[ v_8 < v_9 \implies -\xi_0 + \xi_1 < c_{8,9}, \text{ where } c_{8,9} = 2(b_{7,9} - b_{6,8}) - 2(b_{5,7} - b_{4,6}) + 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2}) \]
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

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

Boundary Constraints:

\[ \alpha < v_0 \Rightarrow \xi_0 < c_{\alpha,0} \text{ , where } c_{\alpha,0} = b_{0,2} - \alpha \]
\[ v_9 < \beta \Rightarrow \xi_1 > c_{9,\beta} \text{ , where } c_{9,\beta} = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \beta \]

Geometric Characterization
“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: $v'' = (v''_0, \ldots, v''_{n-1})$
Reconstruction Error between $v', v''$

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

**APPROXIMATE RECONSTRUCTION**

Reconstruction Error between \( v', v'' \)

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

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

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

Reconstructed Values of 1-31 Oct. Dataset
EVALUATION
ORDERED & UNORDERED RESPONSES

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!
EVALUATION

ORDERED & UNORDERED RESPONSES

1-5 October

1-15 October

1-31 October

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

#### UNORDERED RESPONSES

<table>
<thead>
<tr>
<th>Date</th>
<th>Diameter</th>
<th>m</th>
<th>n</th>
<th>Error-1D</th>
<th>Error-2D</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5 October, $m = 25 \cdot 10^6$, $n = 46$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.02%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>1.1</td>
<td>3.6</td>
<td>1.1</td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>$k = 5$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.03%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>18.3</td>
<td>17.9</td>
<td>5.7</td>
<td>1.6</td>
<td></td>
<td>5.0</td>
</tr>
<tr>
<td>$k = 8$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.1%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>79.9</td>
<td>78.3</td>
<td>16.9</td>
<td>1.4</td>
<td></td>
<td>7.4</td>
</tr>
<tr>
<td>1-15 October, $m = 70 \cdot 10^6$, $n = 79$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.01%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>0.8</td>
<td>1.8</td>
<td>0.7</td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>$k = 5$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.011%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>6.6</td>
<td>6.0</td>
<td>1.9</td>
<td>0.6</td>
<td></td>
<td>2.5</td>
</tr>
<tr>
<td>$k = 8$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.015%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>15.4</td>
<td>14.6</td>
<td>2.5</td>
<td>0.6</td>
<td></td>
<td>2.9</td>
</tr>
<tr>
<td>1-31 October, $m = 250 \cdot 10^6$, $n = 183$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.006%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td></td>
<td>1.4</td>
</tr>
<tr>
<td>$k = 5$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.008%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>5.0</td>
<td>1.4</td>
<td>0.3</td>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td>$k = 8$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.008%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>12.8</td>
<td>11.6</td>
<td>1.4</td>
<td>0.3</td>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td>1-5 October, $m = 800 \cdot 10^6$, $n = 46$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k = 2$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.003%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>1.7</td>
<td>0.5</td>
<td>0.1</td>
<td></td>
<td>0.9</td>
</tr>
<tr>
<td>$k = 5$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.02%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>18.3</td>
<td>18.3</td>
<td>3.4</td>
<td>0.2</td>
<td></td>
<td>2.9</td>
</tr>
<tr>
<td>$k = 8$</td>
<td>exact</td>
<td>est</td>
<td>avg</td>
<td>std</td>
<td>0.009%</td>
<td>max</td>
</tr>
<tr>
<td></td>
<td>79.9</td>
<td>79.5</td>
<td>14.6</td>
<td>0.15</td>
<td></td>
<td>6.5</td>
</tr>
</tbody>
</table>