Authentication of Spatial Queries in Both Vector Spaces and Spatial Networks

Jeff Ku
Auburn University

Introduction

- Spatial data explosion
Problem definition

Query integrity: a mobile user can check that the query result is **authentic, correct and complete.**
Threat Model

• Not only the Cloud might tamper with the data out of its own interests
  – Favor restaurants which pay more on ads

• But also ...
  – Cloud service provider under attack
  – Compromised communication channel

A Cloud You Can Trust?
General Approach Overview

- Owner signs the data with her private key
- Signature and public key are accessible to all clients
- SP returns result and VO for every query
- Clients verify the result using VO, the public key and signature

Related work

- Merkle Hash tree (*CRYPTO’89*)
- VKD-tree, VR-tree (*DBSec’06*)
- MR- and MR*-tree (*ICDE’08, VLDBJ’09*)

- Insert extra information to internal/leaf nodes
- Change existing search algorithms to construct proof
- Significant overhead in the proof (verification object (VO)) for small queries
  -- $O(\log n)$ proof size and verification time

An MR-tree with height = 5, and fan-out (# of entries in each node) = 50
Querying for 1 object needs to return $49 + 5 \times 49 = 294$ additional objects
VN-Auth Verification Algorithm

Preliminaries: Voronoi Diagrams

- Given a set of spatial objects, a Voronoi diagram *uniquely* partitions the space into disjoint regions (cells).
- All the points in $VC(p_1)$ are closer to $p_1$ than to any other object $p'$.

**Voronoi Diagram**

- **Dataset:** Points
- **Distance $D(.,.)$:** Euclidean ($L_2$)

**Voronoi Cell of $p_1$:** $VC(p_1)$

Point $q \subseteq VC(p_1)$

$\Leftrightarrow$ $D(q, p_1) < D(q, p')$

1st NN $\rightarrow p_1$
Preliminaries: Voronoi Diagrams

Lemma: 2nd NN of q is one of Voronoi neighbors of the 1st NN of q.

Lemma: kth NN of q is a Voronoi neighbor of one of 1st, 2nd,..., k-1th NN of q.

Authenticated Voronoi cell

- A standard R-tree index built on the DB
- Every entry at leaf points to an authenticated Voronoi cell
Authenticated Voronoi cell

- No reduction to the fan-out of the R-tree
- No change to existing R-tree-based search algorithms
- High initialization cost & storage overhead
- Query transmission overhead due to multiple signatures

---

VoR-Tree (VLDB’10)

---

Reduce VO size using signature aggregation

- Query result
  
  \[
  \{ p_1, \text{VN}(p_1), \text{sign}(p_1 | \text{VN}(p_1)) \} \\
  \{ p_5, \text{VN}(p_5), \text{sign}(p_5 | \text{VN}(p_5)) \} \\
  \{ p_6, \text{VN}(p_6), \text{sign}(p_6 | \text{VN}(p_6)) \}
  \]

- Verification Object
  
  \[
  \{ \text{A-signature}, p_1, \text{VN}(p_1), p_5, \text{VN}(p_5), p_6, \text{VN}(p_6) \}
  \]
Examples Verifying $k$NN Queries

Example: Verify 1NN

My nearest neighbor is?

1. Verify the signature
2. Check $q \in VC(p_1)$
   - no, $q$ is not in $VC(p_1)$
   - $p_1$ is not the NN
Example: Verify $k$NN ($k=3$)

What are my 3NNs?

$p_1$, $VN(p_1)$, $p_2$, $VN(p_2)$, $p_4$, $VN(p_4)$

Result from the SP

Verified objects

Lemma: 2nd NN of $q$ is one of Voronoi neighbors of the 1st NN of $q$.

2nd and 3rd NN candidate: $(p_2, 3)$, $(p_3, 5)$
Example: Verify $k$NN ($k=3$)

Lemma: $k$th NN of $q$ is Voronoi neighbor of one of $1^{st}$, $2^{nd}$, ..., $k$-1$^{th}$ NN of $q$.

3$^{rd}$ NN candidate: $(p_3, 5)$  
$(p_4, 12)$

Discussion

- The size of the proof matters  
  - Transmission  
  - Verification  
- Easy to adopt: no change to existing index or search algorithm  
- Incremental verification process  
  - In the example: although the 3$^{rd}$ NN is incorrect, the first 2NNs can still be verified  
- Query verification on advanced spatial query types in the paper: $RkNN$, $kANN$, $Spatial Skyline$, etc.
Verify Queries on Road Networks

- Real-world distances are road network distances.
- Objects which are close in the Euclidean space might be far away in reality.

Revisit Threat Model

- Not only the Cloud might tamper with the data out of its own interests
  - Favor restaurants which pay more on ads
- But also ...
  - Cloud service provider under attack
  - Compromised communication channel
- SP might return suboptimal results by applying flawed or inferior algorithms
kNN Results from Online Maps

Euclidean vs. Network Distances

Percent error = 39.7%
Challenges

- Network distance is expensive to compute
- Voronoi cell depends on POIs, as well as road segments
- Cannot assume the verifier has the knowledge of the whole road network
- Need to verify not only the distance, but also the path to each kNN result

Network Voronoi Diagram (NVD)
### Differences between Euclidean and Network Voronoi Diagrams

<table>
<thead>
<tr>
<th></th>
<th>Euclidean VD</th>
<th>Road Network VD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voronoi Polygon</td>
<td>Non-overlapping</td>
<td>Overlapping</td>
</tr>
<tr>
<td>Inside VP</td>
<td>Continuous region</td>
<td>Road segments</td>
</tr>
<tr>
<td>Neighbors share</td>
<td>Edges</td>
<td>Border points</td>
</tr>
<tr>
<td>Distance</td>
<td>Euclidean distances</td>
<td>Road network distances</td>
</tr>
<tr>
<td>Path</td>
<td>Straight lines</td>
<td>Network shortest paths</td>
</tr>
</tbody>
</table>

---

### Preliminaries: Using Voronoi Diagrams to solve kNN query

**Lemma I:** The $2^{nd}$ NN of $q$ is one of the Voronoi neighbors of the $1^{st}$ NN of $q$.

**Lemma II:** The $k^{th}$ NN of $q$ is Voronoi neighbor of one of the $1^{st}$, $2^{nd}$, ..., $k-1^{th}$ NN of $q$.

---

![Diagram of Voronoi diagrams showing distances and nearest neighbors](image)
Lemma 3: The shortest path from $q$ to the first NN $p$ is inside $V(p)$.

Lemma 4: The shortest path from $q$ to the $k^{th}$ NN $p_k$ is inside the union of $V(p_1)$, $V(p_2)$, ..., $V(p_k)$.

Authenticated Network Voronoi Cell

$AVC(p_i) = \{ p_i, VN(p_i), \neu{sign}(p_i | VN(p_i)) \}$

$ANVC(p_i) = \{ p_i, V(p_i), VN(p_i), \neu{sign}(p_i | V(p_i) \cup VN(p_i)) \}$
Examples of Verifying Network $k$NN Queries

Verify Network 1NN Result

\[ \text{Check } q \in V(p_1) \ ? \]
Verify Network 2NN Result

\[ d_n(q, p_2) = \min( d_n(q, b_6) + d_n(b_6, p_1), \]
\[ d_n(q, b_7) + d_n(b_7, p_1) ) \]

Too many network distances to compute

Euclidean Restriction

Euclidean distance:
\[ d(q, p_3) = 3 \]
\[ d(q, p_4) = 6 \]
\[ d(q, p_5) = 7 \]
\[ d(q, p_6) = 9 \]

\[ d_n(q, p_2) = \min( d_n(q, b_6) + d_n(b_6, p_1), \]
\[ d_n(q, b_7) + d_n(b_7, p_1) ) \]
\[ = 5 \]

Computing road network distance is expensive with large \( k \)
Distance Pre-computation

- Generator to border point
- Border point to border point

\[
\text{ANVC}(p_1) = \{ p_1, V(p_1), VN(p_1), DT(p_1), \text{sign}(p_1 | V(p_1) | VN(p_1) | DT(p_1)) \}
\]
Experiments

- Real-world datasets (data points)
  - Sequoia dataset (62,556 points from California-CA)
  - Tiger dataset (569,120 points from North America-NA)
- Server: Windows
- Clients: Motorola Droid mobile phones (WiFi)
- Compared with MR- and MR*-tree (*ICDE’08, *VLDBJ’09)
- R-tree page size: 4kB
- Point: 32-bit; MBR: 64-bit
- Hash/digest: 32-bit, RSA signature: 128-bit

Size of VO v.s. k

![Graph showing the size of VO v.s. k for CA and NA](image)
Page access v.s. $k$

Verification cost v.s. $k$
Range query: size of VO

Conclusions

- No change to spatial indexes and existing search algorithms
- VO only contains objects local to the query
- Verification concerns local regions only
  - VN-Auth is better for small queries, e.g. LBS
  - MR- and MR*-tree are better for large queries
References


Comments and questions?