Contour Map for Event Detection
In Sensor Networks

Jun Yang
CPS 296.1, Spring 2007
Sensor Data Processing
With contents from W. Xue

Announcements (Feb. 1)
- The first round of presentations will be finalized by the end of this week
- Course project milestone 1: March 1
- Reading and review for next Tuesday: Model-Driven Processing by Deshpande et al.

Motivation
- Events in real world (e.g., gas leaks in coal mines) are complex spatiotemporal patterns
- Simple selections not sufficient
  - E.g., \( \text{SELECT} \ldots \text{WHERE gas\_density} > 0.2 \)
- Simple aggregates not sufficient
  - E.g., \( \text{SELECT MAX(gas\_density)} \) FROM ...

Approach
- Base station periodically computes a contour map for each value being measured over the entire space
  - Hopefully more efficiently than collecting all readings
- Base station supports complex event detection queries over the history of contour maps
  - E.g., \[ \text{SELECT alarm()} \text{from contour\_map(temp, 0.1, 0.5)} \text{where pyramid(c.snapshot, "gas\_leakage.xml") sample period 2 min; \}
  - User-defined predicates

What exactly is a contour map?
- Partition space into regions
- Each region spans an area with "similar" readings
- Typically, we partition the domain of all possible values into a number of non-overlapping ranges
- Each region corresponds to one of these ranges, and covers all readings that fall within the range

Contour map for event detection
A slightly more liberal definition
- Not your typical contours!
- Used by the paper in computing contour maps
- Post-processing required to convert them back to standard ones
- Within each region, all readings can be approximated by a single function (of location)
  - Example: a linear regression function \( f(x, y) = c_0 + c_1 x + c_2 y \) predicts the reading at \((x, y)\)
- Different regions are approximated by different functions
- *What's the point of this alternative definition?*
  - Smoothens noisy readings
  - Allows the map to summarize the space more succinctly

Why a contour map?
- Contour maps can be represented more concisely
  - For each region we only need
    - A function (3 parameters for linear regression)
    - A bound on the average error
    - Boundary
- Contours are amenable to compression as needed
  - Merge regions
  - Compress boundary descriptions
- Contour maps can be computed in-network just like an aggregate function
  - Easily ODI

In-network computation
- Benefit: result region requires fewer bits to represent than original regions combined
- Controlled by two user-specified parameters
  - Max average error (within any region): \( \varepsilon \)
  - Max area of any region (as a fraction of the total area): \( p \)
- Merge only if resulting region has
  - Area bounded by \( p \)
  - Average error bounded by \( \varepsilon \)
  - Size (of representation) smaller than the original regions combined
- Greedily merge pairs with highest benefit/cost (size reduction/error bound) ratio

Merging regions
- Other tricks
  - Compressing boundary descriptions
    - Shared boundaries don’t need to be stored twice
    - With orthogonal polygonal regions, only store every other boundary point
  - Snooping
    - Don’t transmit a region if it is subsumed by another one transmitted by a neighbor
      - *Is this spatial suppression?*
      - *Seems to be at odds with the motivation for redundancy*

Incremental computation
- Each node compares its new partial map with one from the last epoch
- If a region’s boundary stays the same, do not retransmit boundary
  - Same applies to function parameters
  - For error bound, do not retransmit if it decreases
  - What if the boundary changes just a little bit?
- A delta region?
Experiments

- Synthetic dataset extrapolated from real data

![Graph showing Network Traffic Mbps for different settings]

- Merge
- Merge + compression
- Merge + snooping
- Merge + incremental
- Merge + all

- In-network linear-regression contours
- In-network equi-width contours
- Server-side linear regression

Biggest saver seems to be temporal compression!

All uses multi-path routing, so conclusion is that linear regression handles missing values better.

Discussion

- An attempt at supporting complex event detection queries
  - Are user-defined functions a good solution?
  - Is any event-specific optimization pushed down into the network?
- What are the most compact spatial summaries?
  - Why just orthogonal ranges?
  - How about lossy compression of boundary?
- How about taking advantage of temporal redundancy?
  - Can you do better than relying on regions remaining identical?
- Are bounds on average errors sufficient as a quality guarantee?
- To what extent is linear regression hard-coded?

Course roadmap revisited

- Introduction
- Applications
- Sensor network as a database
- Next theme: more principled use of statistical models in sensor data processing