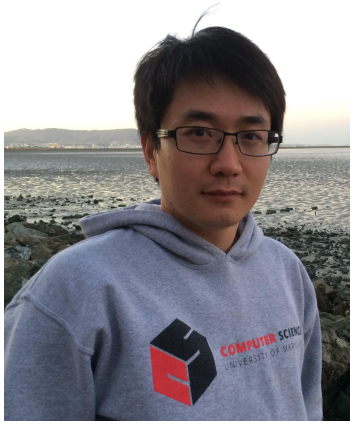
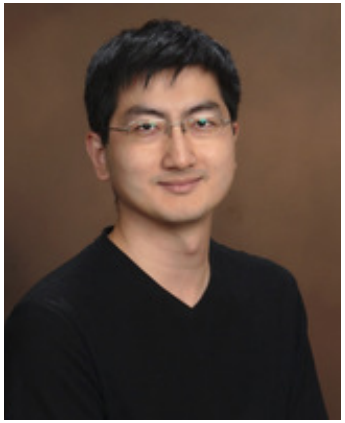
UNIVERSITY OF MARYLAND

Simplification and Refinement for Speedy Spatio-temporal Hot Spot Detection Using Spark

Shangfu Peng



Hong Wei



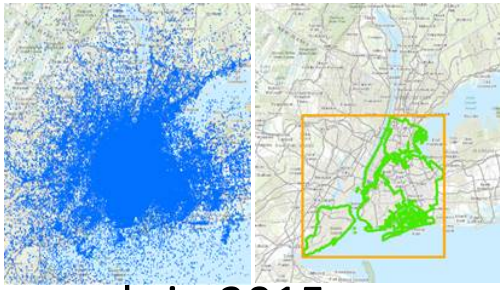
Hao Li

Hanan Samet

{shangfu, hyw, haoli, hjs}@cs.umd.edu
University of Maryland

SIGSPATIAL 2016

Problems



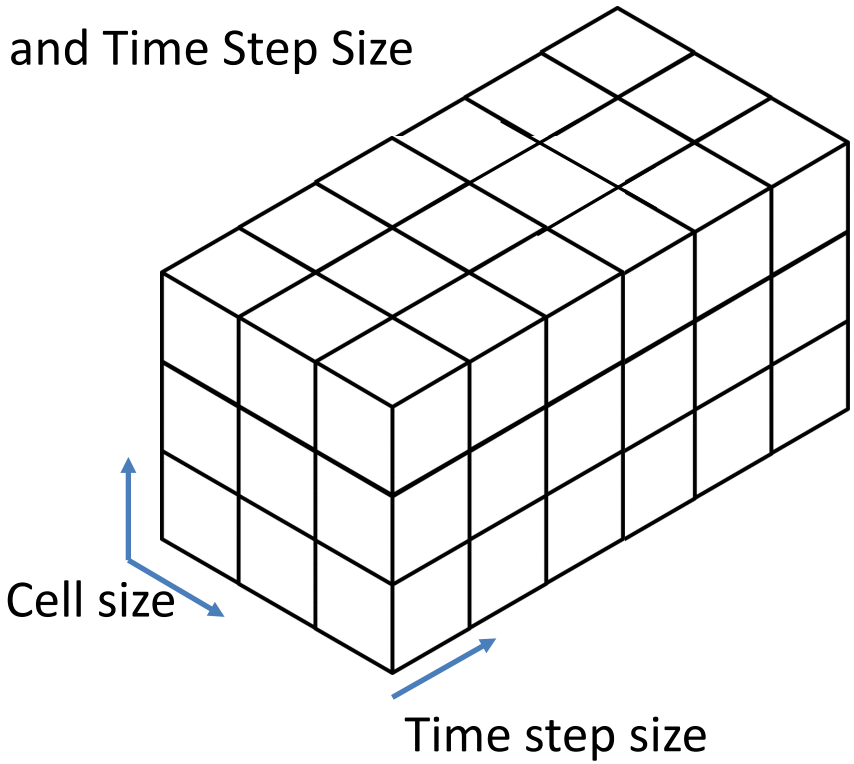
- Identify the top-k spatio-temporal hot spot cells
- Dataset:
  - Drop-off locations in New York City Yellow Cab taxi trip records in 2015.
- Input:
  - Arbitrary Granularity setting: Cell Size and Time Step Size
- Getis-Ord G_i^* statistic:

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S \sqrt{\frac{n \sum_{j=1}^{n} w_{i,j}^2 - (\sum_{j=1}^{n} w_{i,j})^2}{n-1}}}$$

$$w_{i,j} = \begin{cases} 1, & \text{if } c_i \text{ and } c_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases}$$

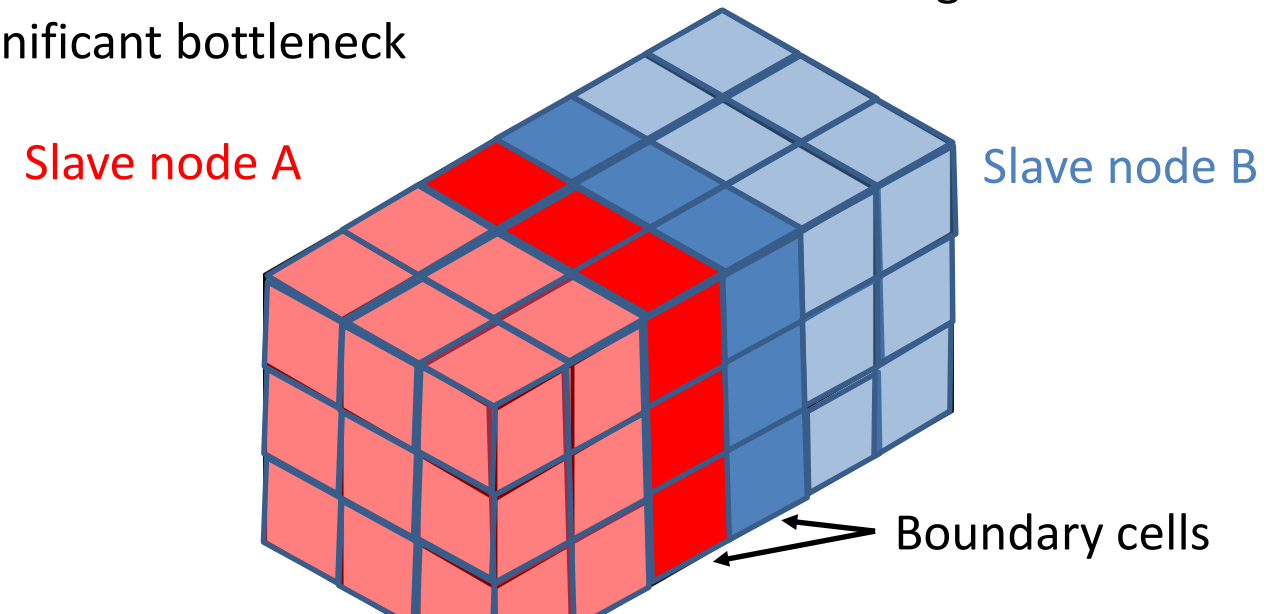
$$\Longrightarrow \sum_{j=1}^{n} w_{i,j} = 27, \forall i$$

$$\Longrightarrow \textbf{compute top-}k\ X_i = \sum_{j=1}^{27} x_{i,j}$$



Challenge

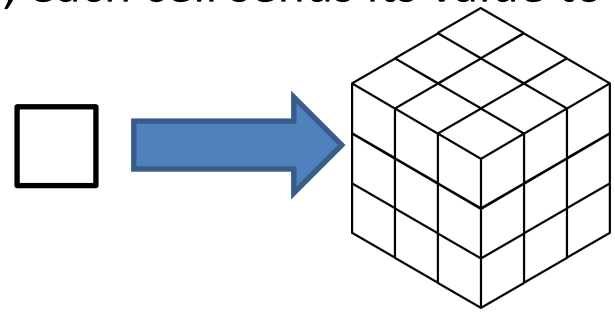
- A Spark cluster?
- Arbitrary fine granularity
  - Any solutions on a coarse granularity take negligible time compared to 18 secs
  - Cells with records are sparse in the cube space in a fine granularity setting
- Efficient partition strategy
  - Every cell needs the values of its neighbor cells, extra network cost
  - **Avoid unnecessary shuffle**: network communication cost during shuffle would become a significant bottleneck



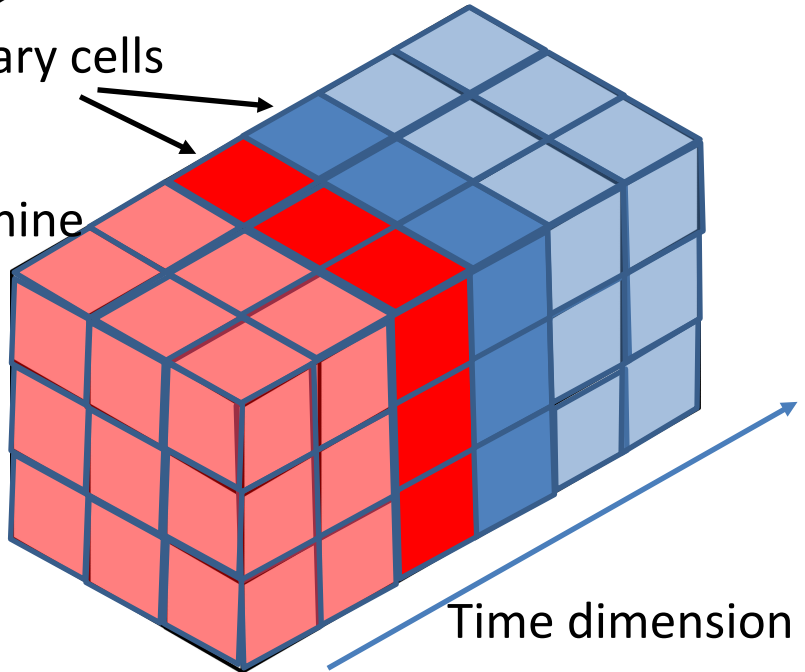
Solutions

- BASIC solution — Easy, but increased the network communication cost a lot
  - Copy each valued cell 27 times
  - Similar to a graph, each cell sends its value to its 27 neighbors

Only valued cell

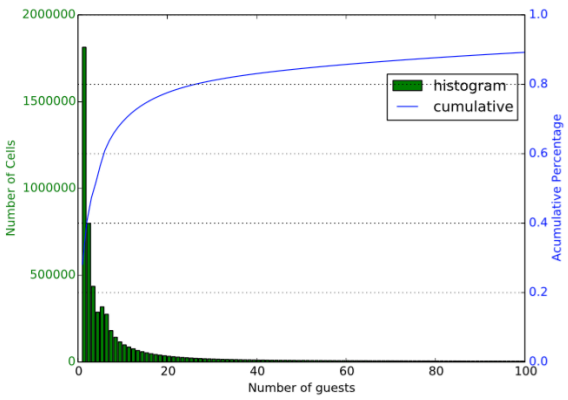


- Low communication solution
  - Partition cells by time dimension
  - Build a HashMap table in each slave machine
  - Compute X_i in each slave machine
  - Recompute X_i for the boundary cells

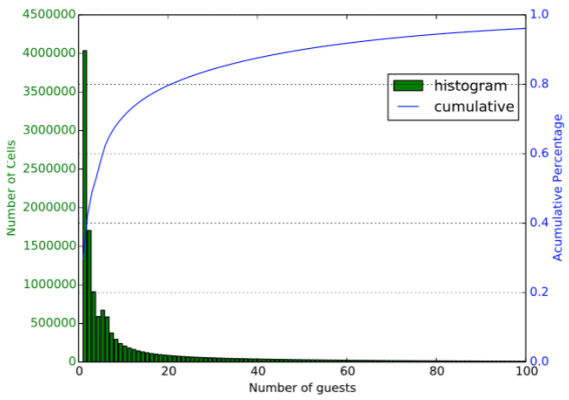


Solution – simplification and refinement (SR)

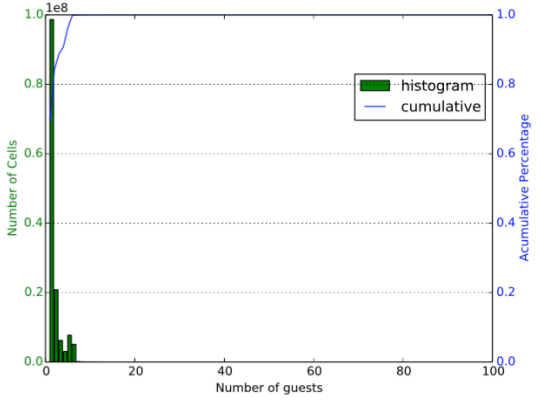
- *Simplification* for speed up and *Refinement* for accuracy
- Intuition for **simplification**
  - Hot spot cells with the highest X_i usually have a large value $x_{i,j}, j \in [1,27]$
  - Cells with small x_i can be thrown away as noise data
  - Keep top *K* valued cells, e.g., *K = 2,000,000* in our submission



(a) $r_s = 0.001, r_t = 1$



(b) $r_s = 0.001, r_t = 0.25$



(c) $r_s = 0.00001, r_t = 0.25$

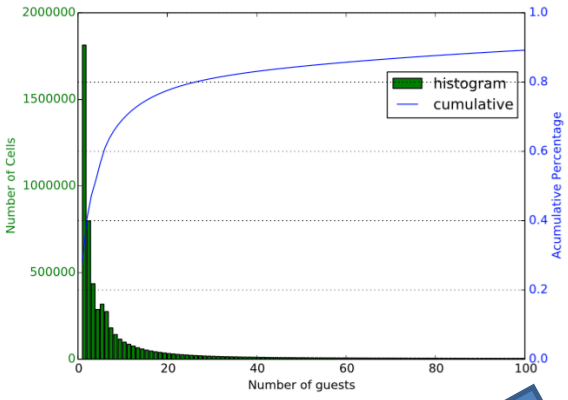
prune $x_i < 10$
cut off 70% cells

prune $x_i < 30$
cut off 85% cells

prune $x_i < 5$
cut off 98% cells

The data follows the long-tail distribution

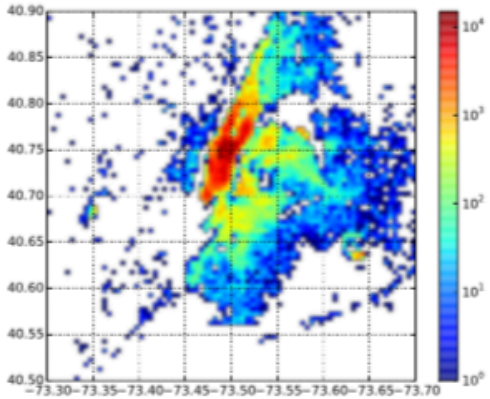
Solution – simplification and refinement (SR)



(a) $r_s = 0.001, r_t = 1$

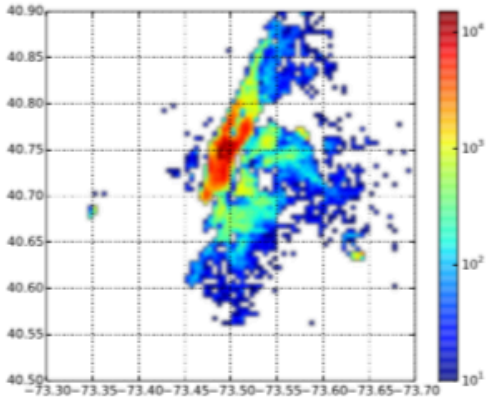
prune $x_i \leq 10$

cut off 70% cells



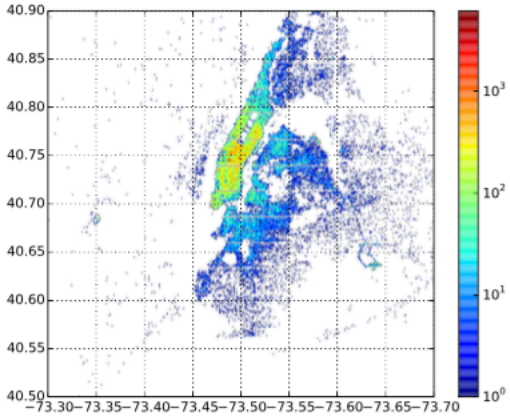
(c) $r_s = 0.005$

cut off 85% cells

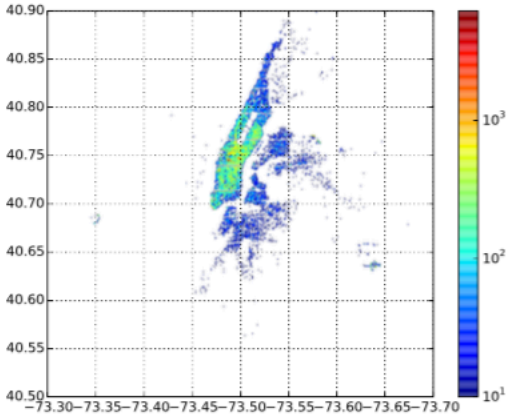


(d) $x_i \geq 10, r_s = 0.005$

cut off 98% cells



(a) $r_s = 0.001$

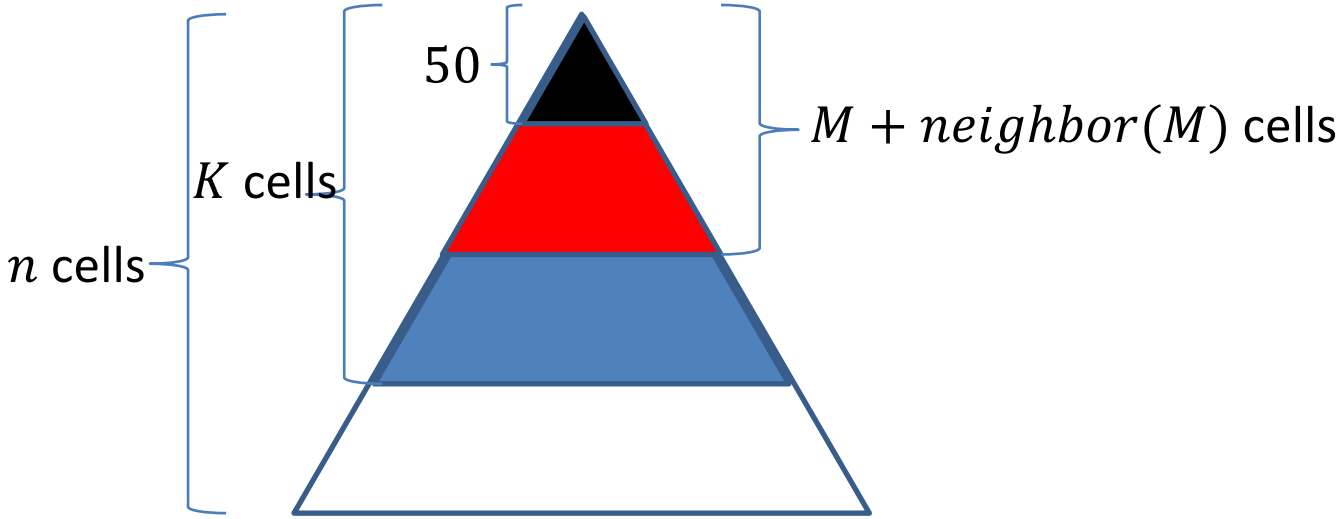


(b) $x_i \geq 10, r_s = 0.001$

Pruning cells with small $x_i < 10$ does not influence the hot spot area

Solution – simplification and refinement (SR)

- Using any solution (BASIC) to compute X_i after simplification
- **Refinement**
  - Using simplification, the top-50 hot spot cells are not accurate
  - Example: a cell that is supposed to rank between 45 to 50 now might fall behind 50 because some of its neighbors were thrown away in simplification
  - Choose the M cells with greatest X_i, e.g., $M = 500$ in our submission,
  - Recompute the exact X_i for the M cells and their neighbor cells



Performance

- Number of drop-off records is 143M
- Preparation (18 – 20 secs)
  - Load all data and cache in memory
  - Process string and timestamp data formats
- Time performance (SR-2M-500 returns the same result)

Table 1: # of Cells in Granularity Settings

Cell Size \ Time Step Size	seven days	one day	six hours
0.1 (10 km)	1275	-	-
0.001(100 meters)	1.93M	6.45M	13.59M
0.00001 (1 meter)	-	-	141.75M

Table 2: Runtime (seconds) without Preparation

Setting \ Solution	SR-2M-500	BASIC	SHIFT
0.1, seven days	≤ 1	≤ 1	5
0.001, seven days	25 - 26	23 - 28	40 - 45
0.001, one day	27 - 29	90 - 120	80 - 85
0.001, six hours	27 - 30	300 - 500	100 - 120
0.00001, six hours	30 - 33	≥ 2000	600 - 700

Conclusions

- Hot spot cells are usually with large x_i or its neighbors with large x_i
- Cells with small x_i can be thrown away as noise data
- Using simplification for speed up and refinement for improving accuracy
- Solving any given granularity setting within 50 secs
- Particularity of this problem (extensions)
  - Require a small number of hot spot cells, e.g., 50 hot spot cells
  - Definition of $w_{i,j}$, e.g., $w_{i,j} = 1$ only when i and j are neighbors
  - Definition of evaluation score, e.g., 2X time loses 16 in score

$$score = 100 * \left(\frac{2 * J(R,S) + (t_R/t_S)}{3} \right)$$

UNIVERSITY OF MARYLAND