UNIVERSITY of HOUSTON
DEPARTMENT OF COMPUTER SCIENCE

A Computational Framework for Finding Interestingness Hotspots in Large Spatio-Temporal Grids

Fatih Akdag and Christoph F. Eick
UH-DMML Group

Abstract

An important task when analyzing large gridded spatial datasets is to find interesting contiguous regions, called *interestingness hotspots*, based on the domain expert's notion of interestingness which is captured in an interestingness function. We present a computational framework which uses a non-clustering approach to obtain interestingness hotspots. A novel hotspot growing algorithm is proposed which grows interestingness hotspots from seed hotspots, and then post-processes the obtained hotspots to remove overlapping hotspots. We claim that our approach is capable of identifying a much broader class of hotspots, which cannot be identified by traditional distance-based clustering algorithms.

Problem

Given

1. Dataset O
2. Neighborhood relation N⊆O×O
3. Interestingness measure i:$2^O \rightarrow \{0\} \cup \Re^+$

The goal of this research is to develop frameworks and algorithms that find, summarize and visualize interestingness hotspots H⊆O; H is an interestingness hotspot with respect to *i,* if the following 2 conditions are met:

1. *i*(H) ≥θ
2. H is contiguous with respect to N; that is, for each pair of objects (o,v) with o,v∈H, there has to be a path from o to v that traverses neighboring objects (w.r.t. N) belonging to H.

In summary, interestingness hotspots H are contiguous regions in space that are interesting (*i*(H) ≥ θ).

Methodology

1. Identify seed regions in the dataset:
   1. Divide dataset into smaller regions of same size called seed candidate regions
   2. Identify seed regions with high interestingness value and grow them in the next step
2. Grow interestingness hotspots from seed regions using a reward function:
   1. Find the neighbor of the seed region which increases a reward function most when added.
   2. Add this neighbor into region and update neighbors list. (A hash set data structure is used to keep neighbors list. Hash set has O(1) runtime complexity for add/remove/contains operations)
   3. Continue adding more neighbors (repeat 1&2) until the reward value cannot be increased for N trials.
3. Post-process obtained hotspots to remove overlaps:
   1. Find the subset of non-overlapping hotspots with the highest total reward value. This problem is equivalent to maximum weight independent set problem in graph theory.

Neighborhood definition:

$$Neighbor(o_1, o_2)$$
$$\Leftrightarrow\ |o_1.x - o_2.x| + |o_1.y - o.y| +\ |o_1.z - o_2.z| + |o_1.t - o_2.t| = 1$$

Reward function: R = interestingness of hotspot x size of hotspot

Hotspot Growing Algorithm

```
FUNCTION AddBestNeighborForRegion(region)

    SET bestNewReward = -1
    SET bestNeighbor = null

    FOREACH neighbor of region
      SET reward = CalculateRewardOf(region + neighbor)
      IF reward > bestNewReward  THEN
          SET bestNewReward = reward
          SET bestNeighbor = neighbor
      ENDIF
    ENDFOREACH

    Add bestNeighbor to region
    Update reward
    Update neighbors list
END FUNCTION
```

Run this function for each seed region as long as there is no more improvement in the reward of the region for a number of iterations.

Post-processing Algorithm

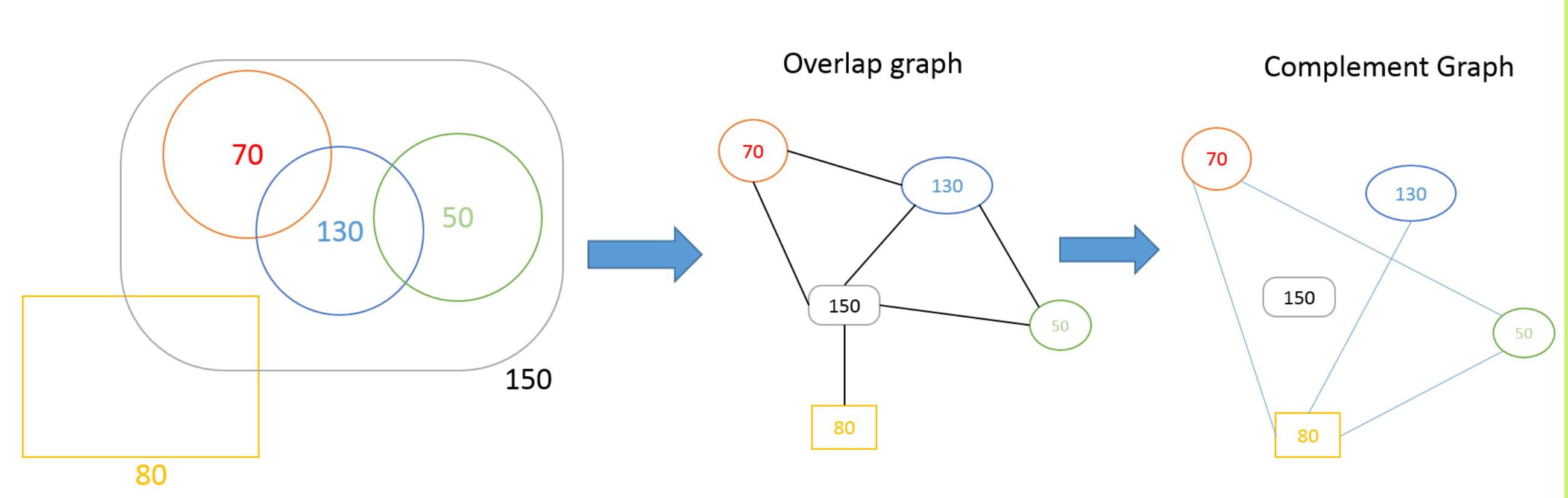
Input: a set of hotspots H, and an overlap threshold λ

Find a subset H'⊆H for which $\sum_{h \in H}$ i(H) is maximal,

subject to the following constraint: ∀h∈H'∀h'∈H' λ≥overlap(h,h')

overlap(h,h') = (number of grid-cell h and h' have in common)/(total number of grid-cells in h and h')

1. Create an undirected graph where each vertex is a hotspot and weight of each vertex is the reward of hotspot.
2. Create an edge between each pair of vertices if they overlap higher than a threshold percentage.
3. Find connected components in the graph.
4. Find the maximum weight independent set in each component.
5. Union of vertices in all maximum independent sets is the result



Interestingness Functions

1.Correlation interestingness function:

$$i_{corr\ (p1,\ p2)}(\mathrm{H}) = \begin{Bmatrix} 0, & if\ |correlation(p_1, p_2)| < \theta \\ |correlation(p_1, p_2)| - \theta, & otherwise \end{Bmatrix}$$

2. *Variance interestingness function:*

$$i_{var\ (p1)}(\mathrm{H}) = \begin{Bmatrix} \theta - variance(p_1), & if\ variance(p_1) < \theta \\ 0, & otherwise \end{Bmatrix}$$

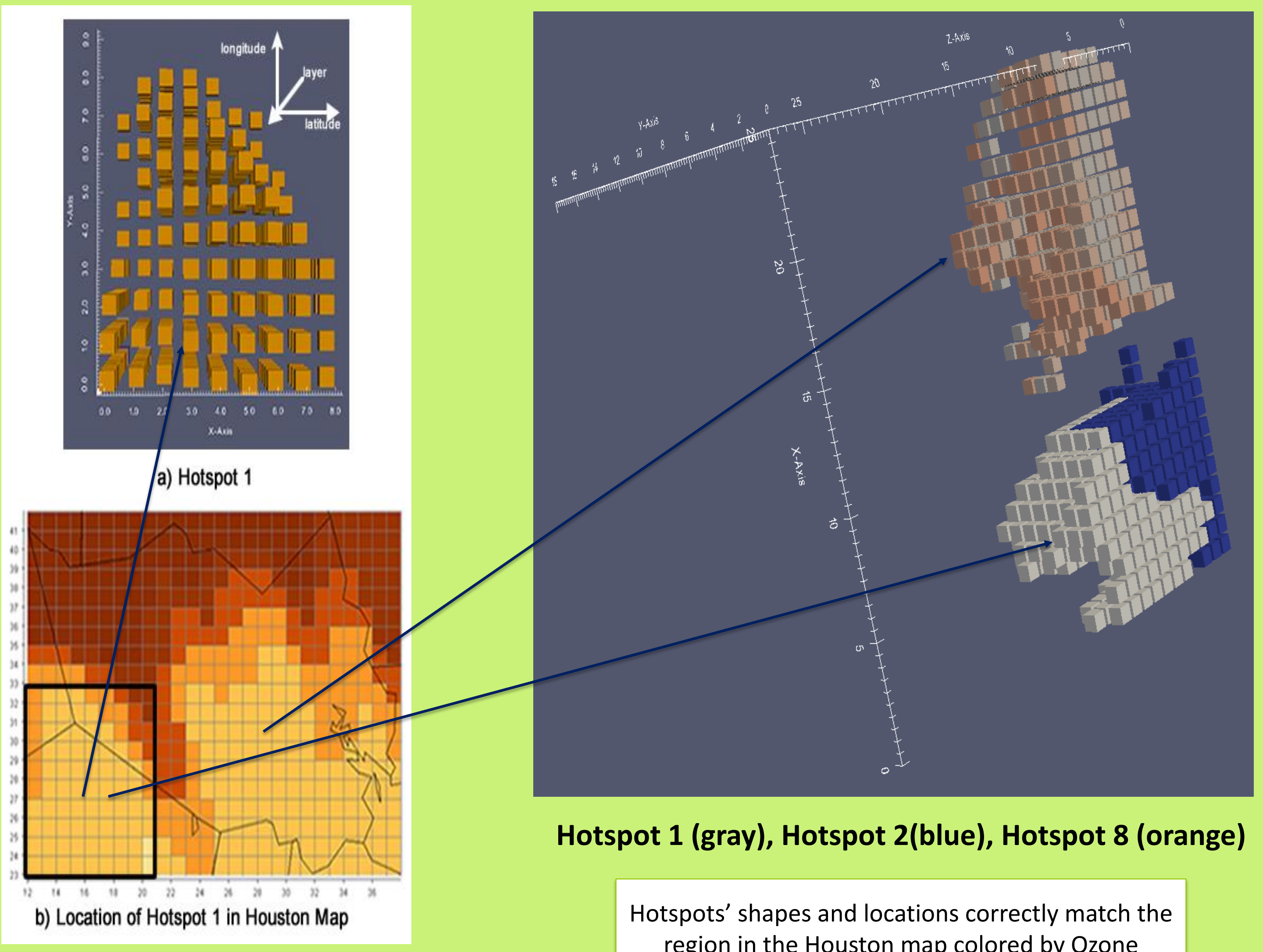
where θ is a threshold value, p_1 and p_2 are attributes.

Experimental Results

- Identify regions with low variance of ozone concentration
- Used a 3D grid for the Houston Metropolitan area for Sept. 1, 2013, noon: 26x19x27=13,338 grid cells

- Reward function: $\varphi(R_i)$ = *interestingness*(R_i) x size(R_i)$^\beta$ where β parameter determines preference for larger regions, where *i* is the variance interestingness function.

- Seed size was set to 3x3x3 (432 seed candidates)
- Use β=1.01
- Variance threshold = 1x10^{-3} ppmV
- Results: 47 hotspots identified
- Many of the hotspots are overlapping



A low variance hotspot and its location in the map

Hotspot 1 (gray), Hotspot 2(blue), Hotspot 8 (orange)

Hotspots' shapes and locations correctly match the region in the Houston map colored by Ozone concentration in each grid.

Each hotspot represents a region with low variation of Ozone.

Conclusion

To the best of our knowledge, proposed hotspot discovery algorithm is the only hotspot discovery algorithm in the literature that grows seed regions using a reward function

We evaluated our framework in a case study and the proposed hotspot discovery algorithm succeeded to find interestingness hotspots by maximizing the plugin reward function.

We claim that the proposed framework is capable of identifying a much broader class of hotspots, compared to other approaches.

We are working on parallel processing of hotspot growing phase and comparing our approach with clustering-based approaches.