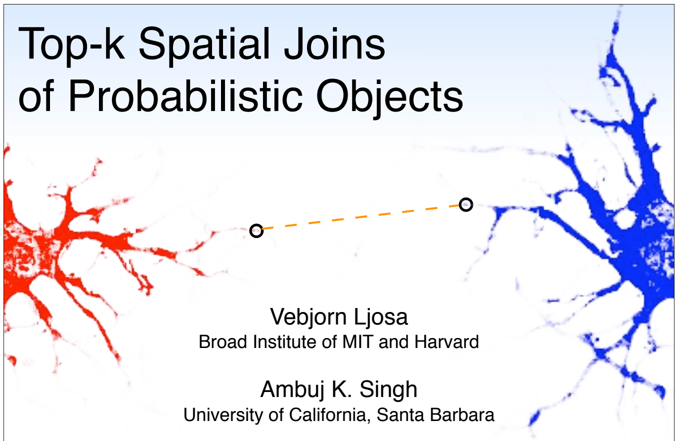
Top-k Spatial Joins of Probabilistic Objects



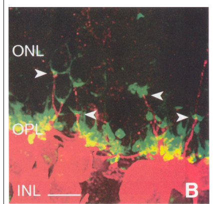
Vebjorn Ljosa
Broad Institute of MIT and Harvard

Ambuj K. Singh
University of California, Santa Barbara

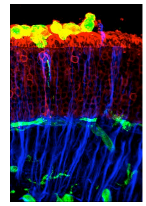
April 9, 2008 , Cancún, México
Geeks Gone Wild: International Conference on Data Engineering (ICDE)

Many biological questions are really spatial joins!

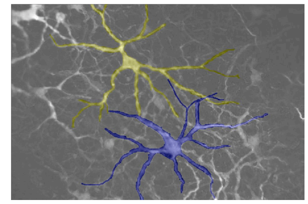
Examples from neuroscience:



Bipolar cells and synaptic terminals



Müller cells and macrophages

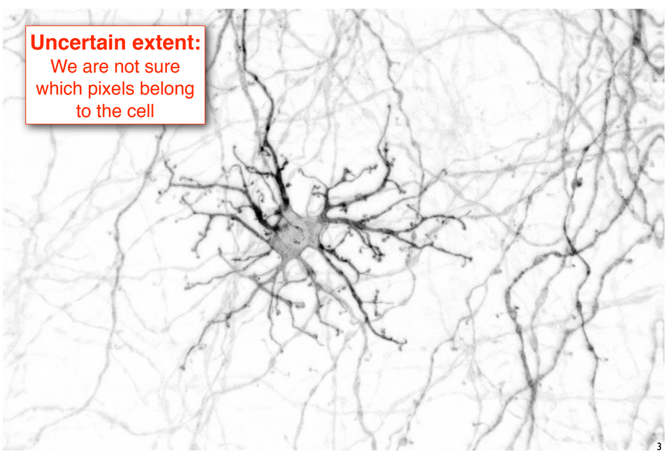


Horizontal cells

[Geoff Lewis; Mark Verardo]

Horizontal cells are hard to segment

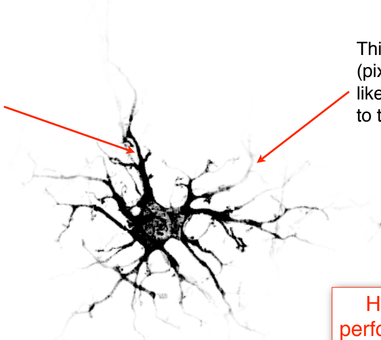
Uncertain extent:
We are not sure which pixels belong to the cell



Probabilistic mask of one horizontal cell

This point (pixel) is more likely to belong to the cell

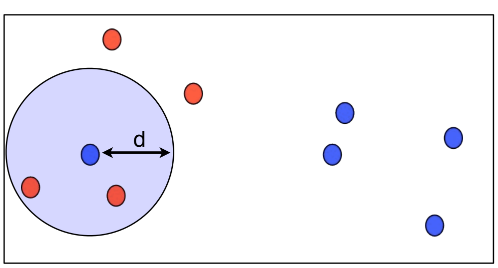
This point (pixel) is less likely to belong to the cell



How do we perform a spatial join of such cells?

[Ljosa and Singh, ICDM 2006]

Spatial joins



Spatial join of objects with certain extent:
Find all pairs of red and blue points less than d apart

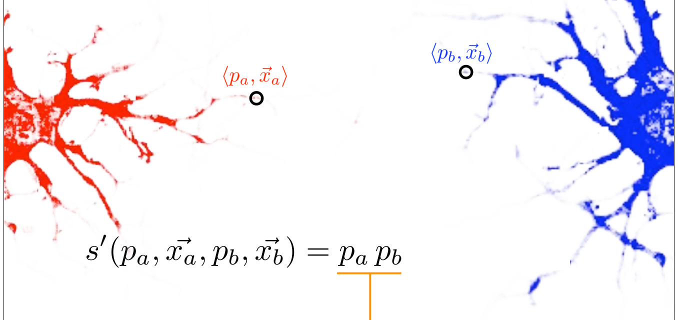
Each ● is either a match or not to ●.

Two types of probabilistic spatial join (PSJ) queries

Threshold PSJ: Given two sets A and B of probabilistic objects, and a score threshold tau, find all pairs (a, b) in A x B such that s(a, b) ≥ tau

Top-k PSJ: Given two sets A and B and a natural number k, find a set R ⊆ A x B of size k such that other pairs in A x B score no higher than the lowest-scoring pair in R.

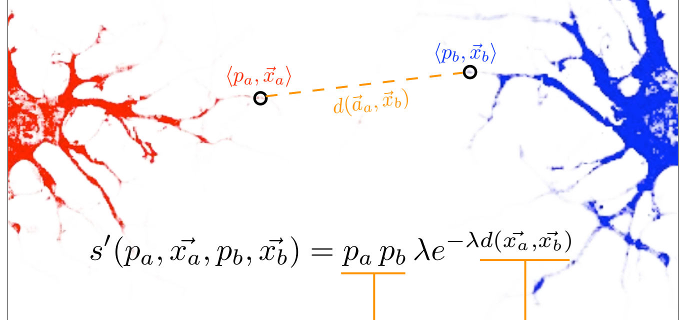
Score function



$$s'(p_a, \vec{x_a}, p_b, \vec{x_b}) = \underbrace{p_a\, p_b}$$

Joint probability that point A belongs to the red cell and point B belongs to the blue cell

Score function



$$s'(p_a, \vec{x_a}, p_b, \vec{x_b}) = \underbrace{p_a\, p_b}\, \lambda \underbrace{e^{-\lambda d(\vec{x_a}, \vec{x_b})}}$$

Joint probability that point A belongs to the red cell and point B belongs to the blue cell

Distance between point A and point B

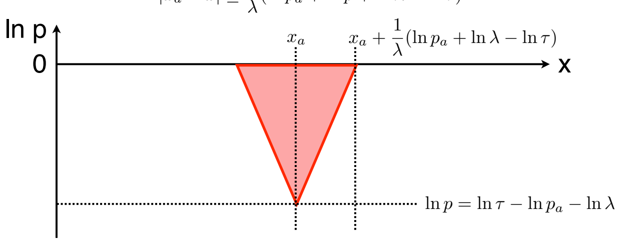
Used in biogeography and ecology [Nekola and White, *J. Biogeography*, 1999]

Threshold PSJ queries

Given a point a = (x_a, p_a) and a threshold tau, where must a point b = (x_b, p_b) be in order to satisfy the query?

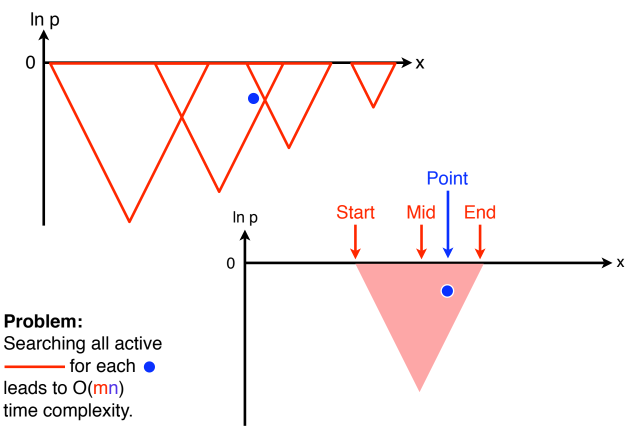
$$p_a\, p_b\, \lambda e^{-\lambda d(\vec{x}_a, \vec{x}_b)} \geq \tau$$

$$|x_a - x| \leq \frac{1}{\lambda}(\ln p_a + \ln p + \ln \lambda - \ln \tau)$$



Very important equivalence: $s' \geq \tau \Leftrightarrow$ ● ∈ ▼

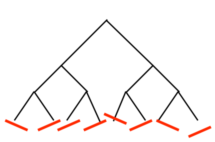
Plane sweep for threshold PSJ queries

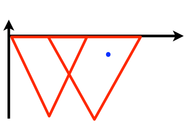


Problem:
Searching all active —— for each ●
leads to O(mn)
time complexity.

Attempt at an efficient solution

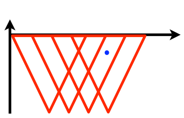
Store all active —— in a balanced, dynamic tree
(e.g., a red-black tree).





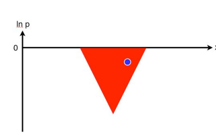
Add events for intersections to keep —— ordered by ln p.

Problem: Could have O(n^2)
intersection events.



A property of the score function saves us

$$|x_a - x_b| \leq \frac{1}{\lambda}(\ln p_a + \ln p + \ln \lambda - \ln \tau)$$



Left edge: $\ln p = -\lambda x + (\ln \tau - \ln p_a - \ln \lambda + \lambda x_a)$

Right edge: $\ln p = \lambda x + (\ln \tau - \ln p_a - \ln \lambda - \lambda x_a)$

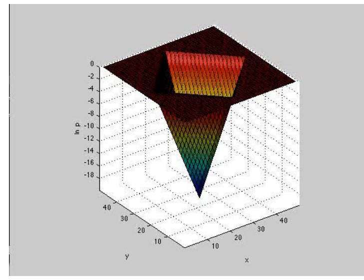
Slope is always λ or –λ

Solution: Use two trees: One for downsloping lines and another for upsloping lines.

Time complexity

- Sort events: O(n log n) + O(m log m)
- There are O(n + m) events
  - Processing a start/mid/end event: O(log n)
  - Processing a point event: O(log n + k')
    - k' is the number of results for this point
- Time complexity: O(m log m + (n + m) log n + k)
  - k is total number of results
  - If we assume that m = n: O(n log n + k)

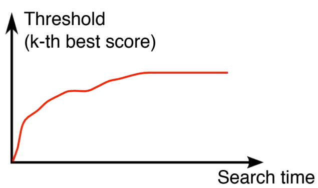
Algorithm generalizes to multiple dimensions

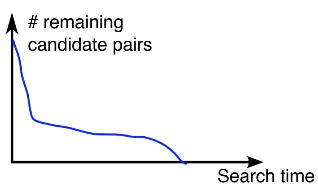


In 2D: Pyramid instead of triangle

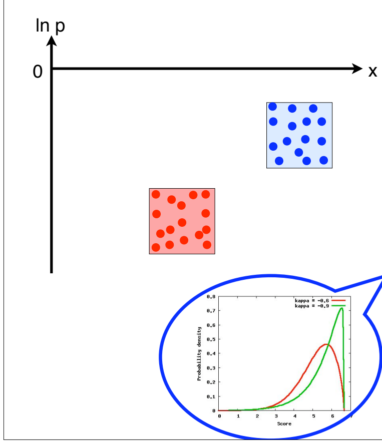
Plane sweep for top-k PSJ

- Query: Find the k top-scoring pairs.
- Plane sweep algorithms adapt easily
  - Move start and end events as threshold increases.
- Key to efficiency is to find some good pairs early
  - Brings the threshold up
  - Prunes most of the dataset





Global scheduling of top-k PSJs



Group r points into MBRs in (x, ln p)-space.

Expected value of s_{max}?
Difficult to compute, and sampling is too slow.

With many points in each box…
Upper bound on score (based on MBR) is a good approximation of s_{max}.

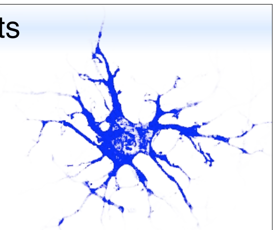
New time complexity:

$$O\left(\frac{nm}{r^2}\left(\log\frac{nm}{r^2} + \rho r \log r + \rho r k\right)\right)$$

Sorting Pruning rate

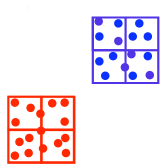
Experiments

Datasets (43k and 52k points) based on horizontal cell images

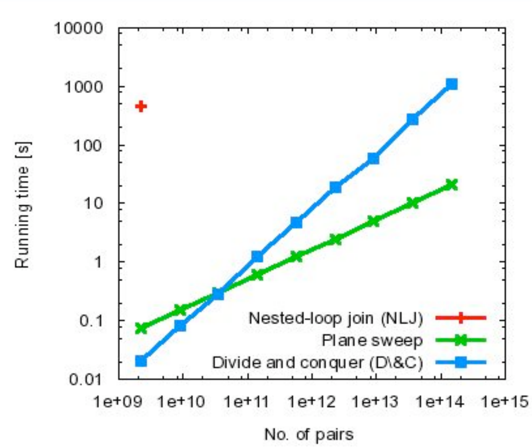


Increased size synthetically (copy & shift) up to 300 times (from 10^9 pairs to 10^{14} pairs)

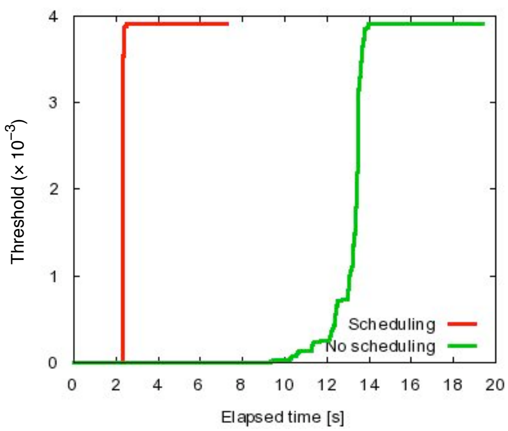
Compared to NLJ and a simple divide-and-conquer technique:
- Split boxes recursively until they
  - can be pruned based on the threshold and s_{max}, or
  - are small enough to be joined with NLJ



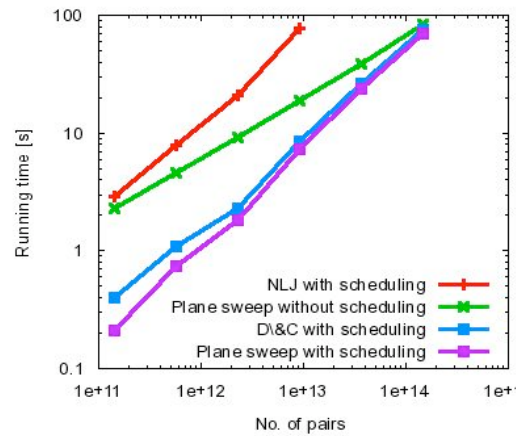
Experiments: Scalability of threshold PSJ queries



Threshold increases faster with scheduling



Scalability of top-k PSJ queries



Conclusion

Probabilistic spatial joins
- Geographical information systems
- Biomedical image analysis

Technically challenging
- Score depends on not only distance, but on both probabilities
- Finding top-ranking results: spatial join and top-k query at once

Efficient algorithms
- Threshold PSJs and top-k PSJs
- Plane sweeps in $O(n \log n + k)$ time
- Global scheduling: faster top-k by finding high-scoring pairs early

Future work
- Efficient algorithms for more than 2 dimensions
- Compare experimentally to Kriegel et al. [DSFAA 2006]

Acknowledgements: This work was supported in part by grant no. ITR-0331697 from the National Science Foundation. Horizontal cell micrographs were provided by Geoffrey P. Lewis from the laboratory of Steven K. Fisher at UCSB.