

A Fast Grid-Based Scan Statistic for Detection of Significant Spatial Disease Clusters

Daniel B. Neill and Andrew W. Moore
Carnegie Mellon University

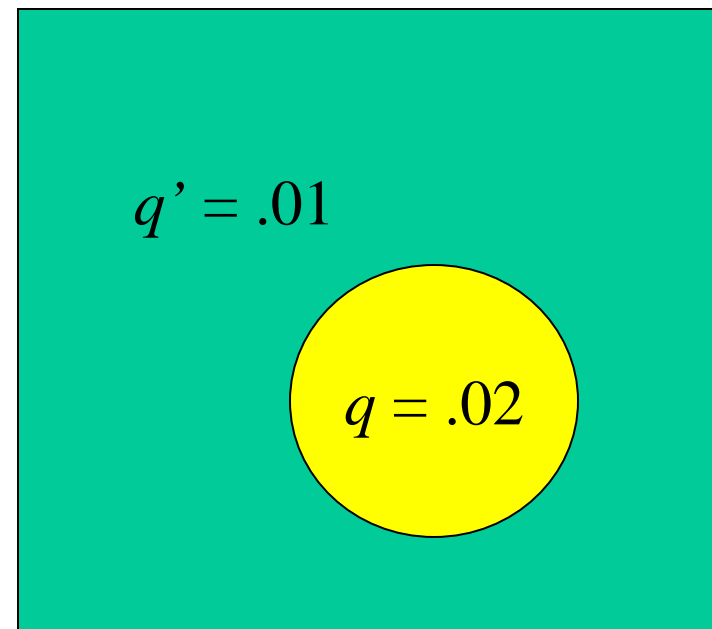
Introduction

- Fast spatial scan statistics are needed to rapidly detect spatial clusters of disease cases.
- For real-time detection of disease outbreaks, a system should be able to find a significant cluster in minutes rather than days.
 - Faster detection can save lives!

Spatial scan statistics

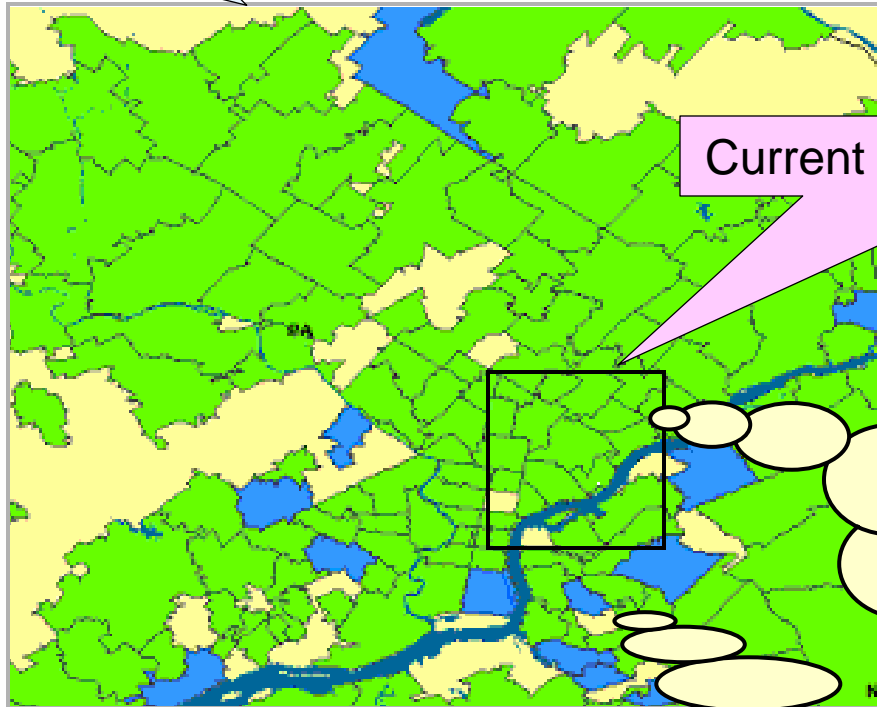
- Kulldorff's spatial scan statistic (1997) is individually most powerful for finding a single region of elevated disease rate.
- Given a region with uniform disease rate q inside the region and $q' < q$ outside, this test is more likely to detect the cluster than any other test (for a fixed probability of Type I error).
- Compute likelihood ratio statistic D_K for each region:

$$D_K = C \log \frac{C}{P} + (C_{tot} - C) \log \frac{C_{tot} - C}{P_{tot} - P} - C_{tot} \log \frac{C_{tot}}{P_{tot}}$$



One Step of Spatial Scan

Entire area being scanned



Current region being considered

I have a population of 5300 of whom 53 are sick (1%)
[Score = 1.4]

Everywhere else has a population of 2,200,000 of whom 20,000 are sick (0.9%)

Scan statistics: computational issues

- Must perform computations over all spatial regions S to find the maximum $D_K(S)$.
- In order to find the statistical significance (p-value) of this region, must generate large number R of replica grids (same underlying populations, but no disease cluster) and calculate statistics over all regions of each replica.
- Complexity grows rapidly with number of data points M : infeasible for large databases!

Grid-based scan statistics

- The standard scan statistic is slow when the number of data points M is large.
- A simple solution is to aggregate points to a uniform $N \times N$ grid: complexity is then a function of N , not M .

P=5000 C=27	P=3500 C=14	P=4500 C=22	P=3000 C=15	P=1000 C=5
P=5000 C=26	P=4000 C=17	P=3000 C=12	P=2000 C=12	P=1000 C=4
P=5000 C=19	P=5008 C=25	P=9000 C=43	P=9000 C=37	P=4000 C=20
P=4800 C=18	P=4800 C=20	P=4000 C=40	P=3000 C=22	P=4000 C=16
P=4700 C=20	P=3000 C=13	P=3000 C=18	P=2000 C=20	P=1000 C=4

Underlying population of square

Number of disease cases in square

This region has an overdensity of disease cases

A naïve grid-based approach

- Search all square regions, return the highest value of the scan statistic, and do randomization testing.
 - Use the “cumulative density” trick!
- This is faster than the standard scan statistic when the grid is dense.
- However, still too slow for real-time detection!

For a 512 x 512 grid, with 1000 replications:
45 billion regions to search!

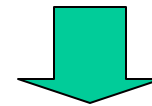
The cumulative density trick

- An old and useful method for fast grid-based computations.
- Calculate matrix of cumulative counts:

$$cc(i, j) = c(i, j) + cc(i-1, j) + cc(i, j-1) - cc(i-1, j-1)$$

- Now we can calculate the total count of any region by adding/subtracting at most four cumulative counts.
 - Middle 4 squares:
 $(42 + 7) - (12 + 20) = 17$
 - Rightmost 3 columns:
 $75 - 28 = 47$

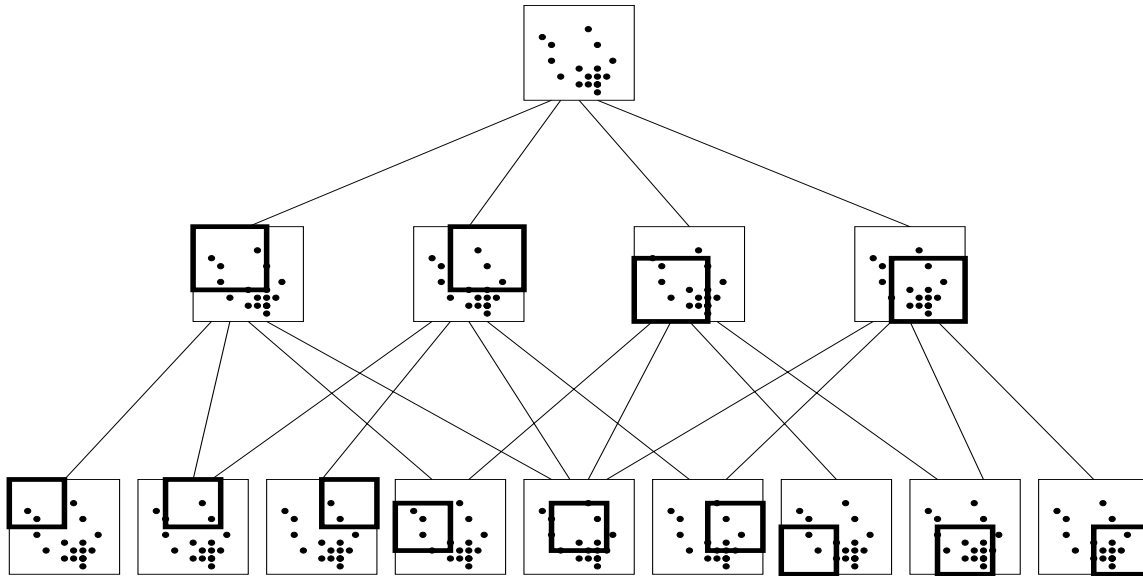
8	5	7	1
9	3	1	2
4	5	8	6
7	3	2	4



28	44	62	75
20	31	42	54
11	19	29	39
7	10	12	16

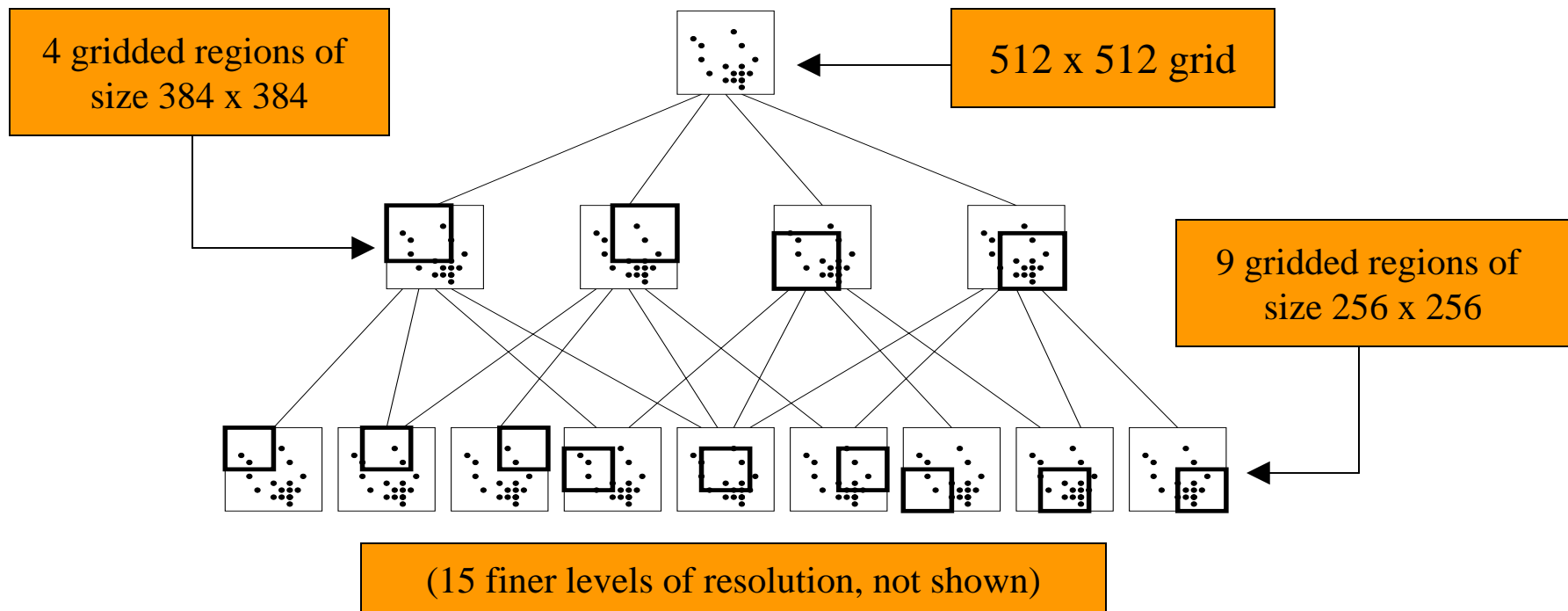
A fast grid-based approach

- We propose a new multi-resolution algorithm which:
 - Partitions the grid into overlapping regions.
 - Performs a top-down search, first at coarse resolutions (large regions) then successively finer resolutions as necessary.
 - Prunes regions which cannot contain the maximum density region.



“Gridded” regions

- For each resolution, we define a set of overlapping regions which cover the entire grid; only a small proportion of regions are “gridded.”
- If we can confine our search to the gridded regions, and search very few non-gridded regions, we can reduce the number of regions searched by a factor of 10,000 or more.



Bounding region density

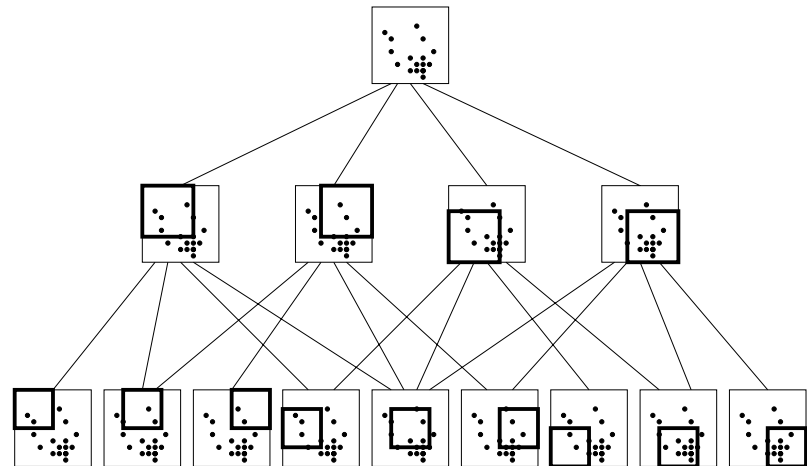
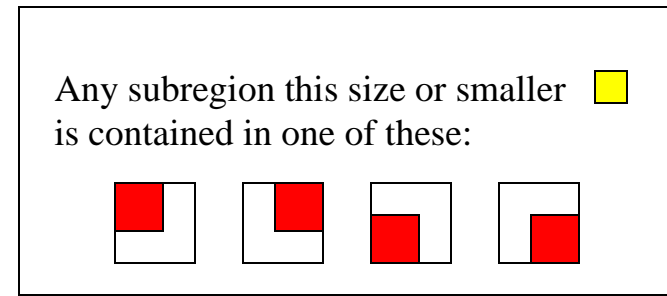
- We precompute bounds on the populations and densities of the squares contained in each gridded region.
- This can be done quickly since there are relatively few gridded regions.
- We can use this information to compute an upper bound on the score D_K for all subregions of a given region.

Region pruning

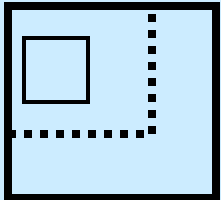
- In our top-down search, we keep track of the best region found so far, and its score.
- If the upper bound for a region is worse than the best score so far, we can prune it.
 - If *no* subregion can be optimal, prune completely (don't search any subregions).
 - If no *large* subregion can be optimal, recurse on the smaller gridded subregions.
 - If neither case applies, we must search gridded and non-gridded subregions.

Why overlapping regions?

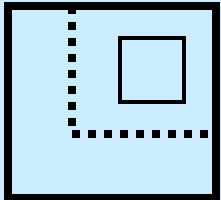
- We must search over all regions, not just gridded regions.
- We can show that any sufficiently small subregion of a region S is contained entirely in a gridded child region of S .
- Thus if we can show that no large subregion of S can be the maximum density region, we can just search recursively on the children of S .
- Without overlap, we would have to show that no subregion can be the maximum density region.
- Thus the use of overlapping regions allows for another, more useful form of pruning.



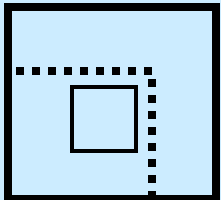
A subregion of me could be one of five types...



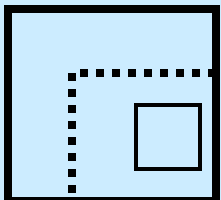
...entirely inside my top left gridded child



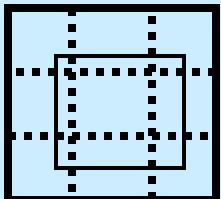
...entirely inside my top right gridded child



...entirely inside my bottom left gridded child

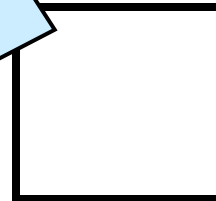


...entirely in my bottom right gridded child



...not entirely inside any of my 4 gridded children

Consider the set of subregions of a gridded region.



FACT: Any subregion of this type must be big...

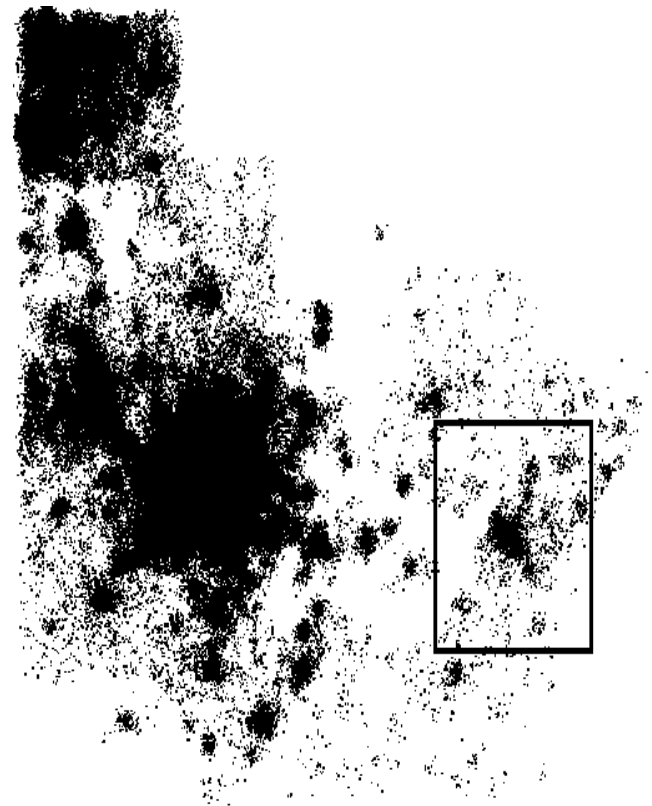
...and we can put fairly tight bounds on how well any region of this type can score

The algorithm

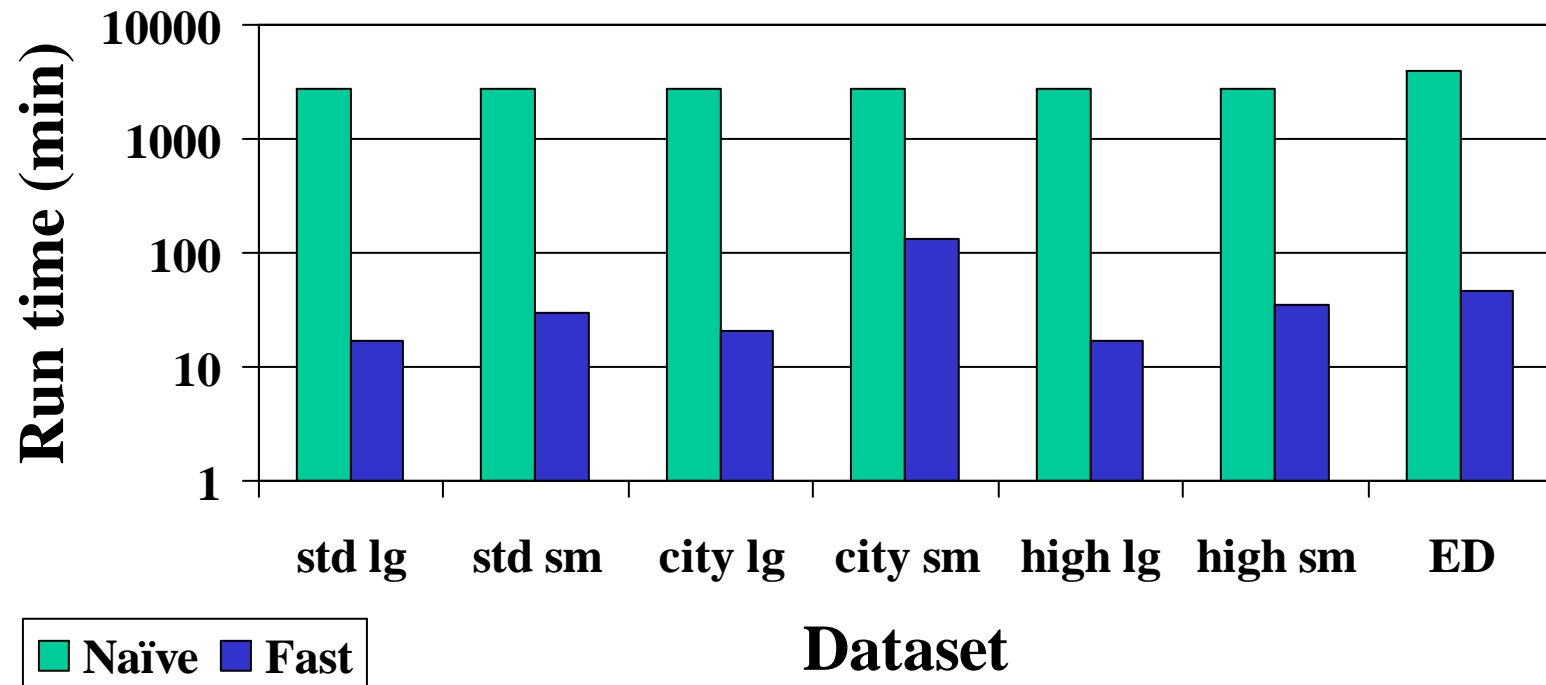
- Top-down, best-first search of gridded regions, followed by top-down, best-first search of non-gridded regions (if necessary).
- Basic step: take best (highest density) region from priority queue, examine, and either prune children or add to queues.
- Mark regions so they will not be searched more than once (multiple parents make this necessary).
- See paper for a more detailed description.

Results: a fast scan statistic

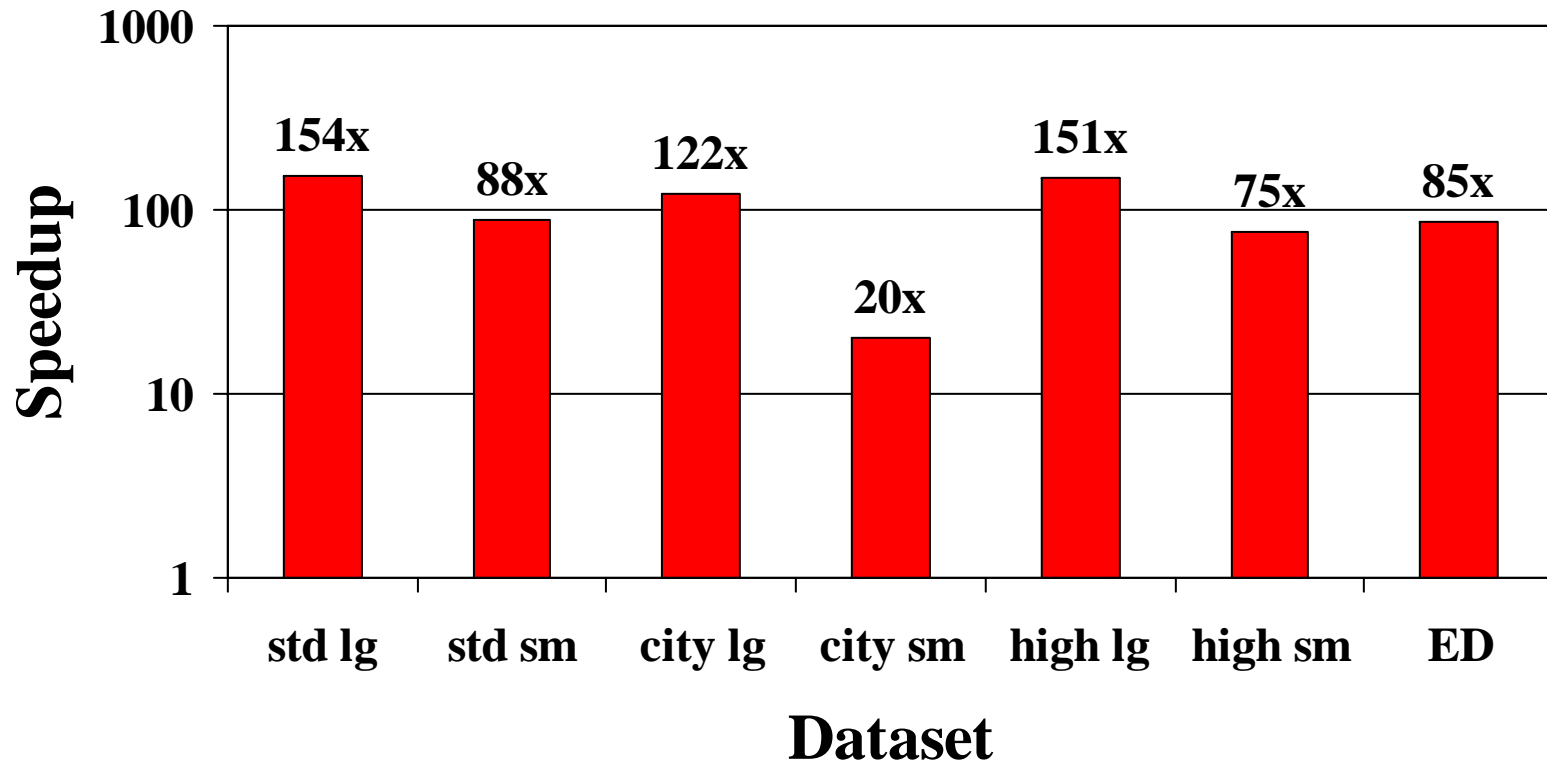
- Theoretical complexity $O(N^2)$ (vs. naive N^3), if maximum density region sufficiently dense.
 - If not, can use several other tricks to speed calculation.
- In practice: speedup 20-150x.
 - Emergency Dept. dataset (600K records): 45 minutes. With naive approach: 66 hours!
 - Similar performance on a variety of artificially generated datasets.



Results (512 x 512 grids)



Speedups (512 x 512 grids)



Conclusions

- Our fast algorithm results in significant speedups on both real and artificial datasets, making real-time detection of disease clusters computationally feasible.
- Our current focus is applying this algorithm to the automatic real-time detection of disease outbreaks, based on regional hospital and national-level pharmacy data.