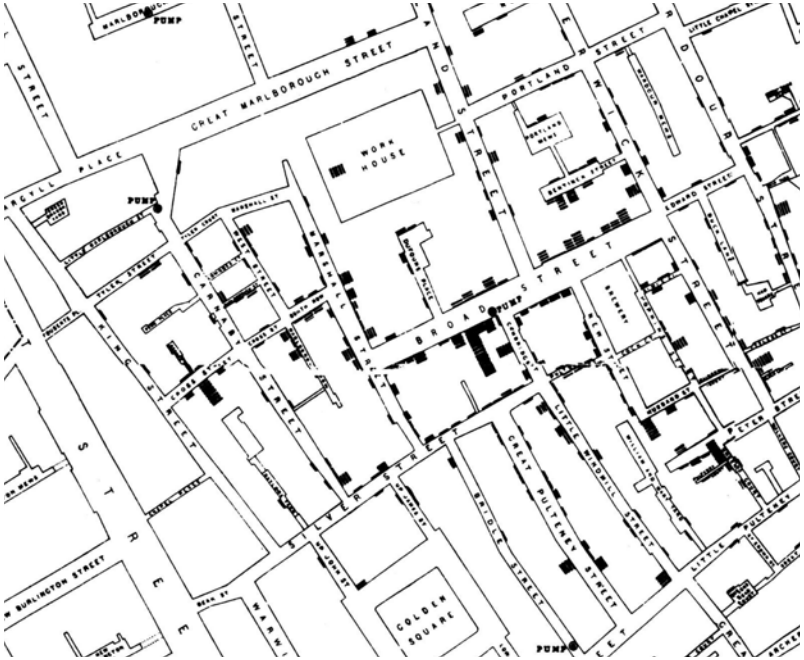
A LRT Framework for Fast Spatial Anomaly Detection*

Mingxi Wu (Oracle Corp.)
Xiuyao Song (Yahoo! Inc.)
Chris Jermaine (Rice U.)
Sanjay Ranka (U. Florida)
John Gums (U. Florida)

* Work undertaken when all authors were with the University of Florida

Spatial Anomaly Detection

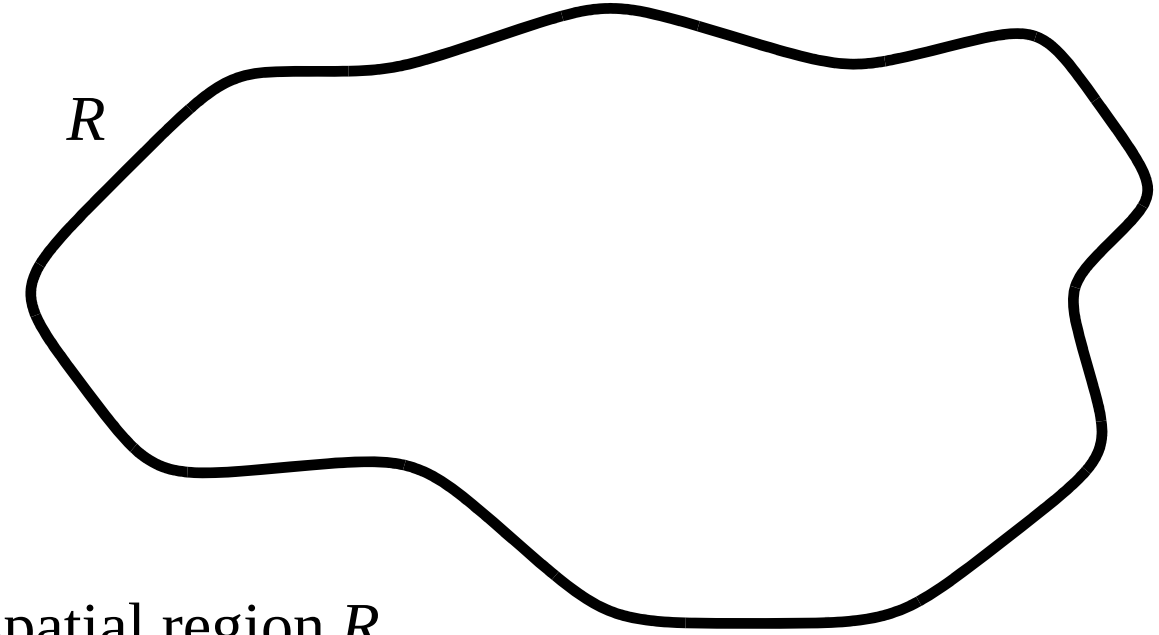
- Long-studied problem in statistics, epidemiology, data mining
- Usual context is finding spatial “hot spots” (disease, sales, etc.)
- Classic example: 1854 Broad Street cholera outbreak



- John Snow plotted all cases and determined a well as the source
- Rebuked the “miasma” theory of disease

Spatial Anomaly Detection

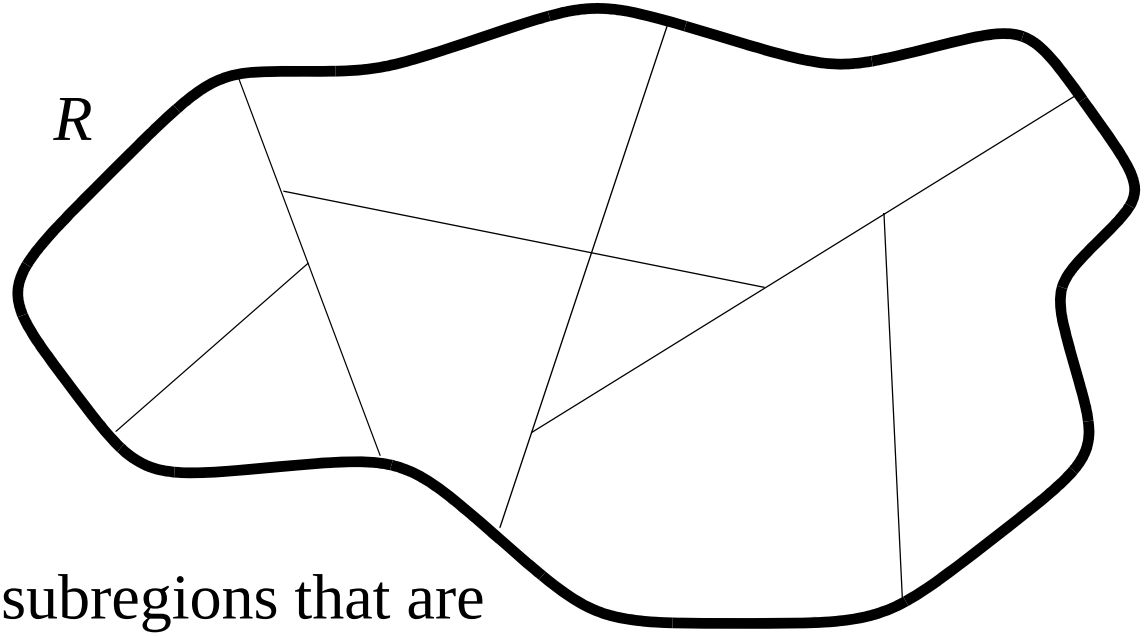
- Underlying statistical model is typically quite simple



Have some spatial region *R*

Spatial Anomaly Detection

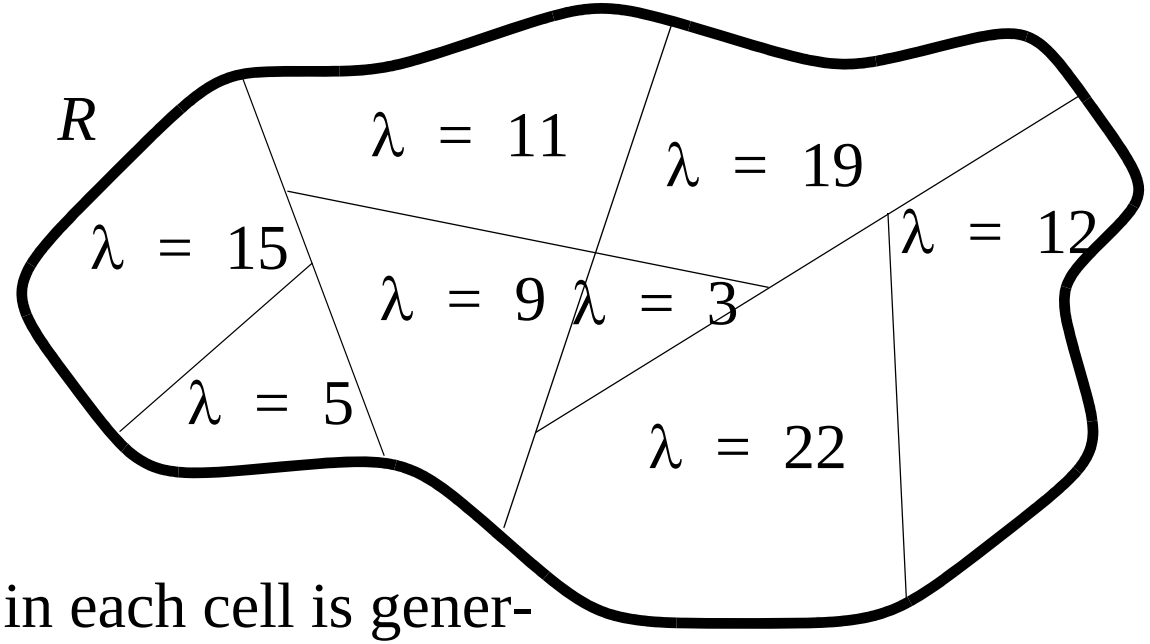
- Underlying statistical model is typically quite simple



Divided into subregions that are small enough that iter-region variation is not interesting

Spatial Anomaly Detection

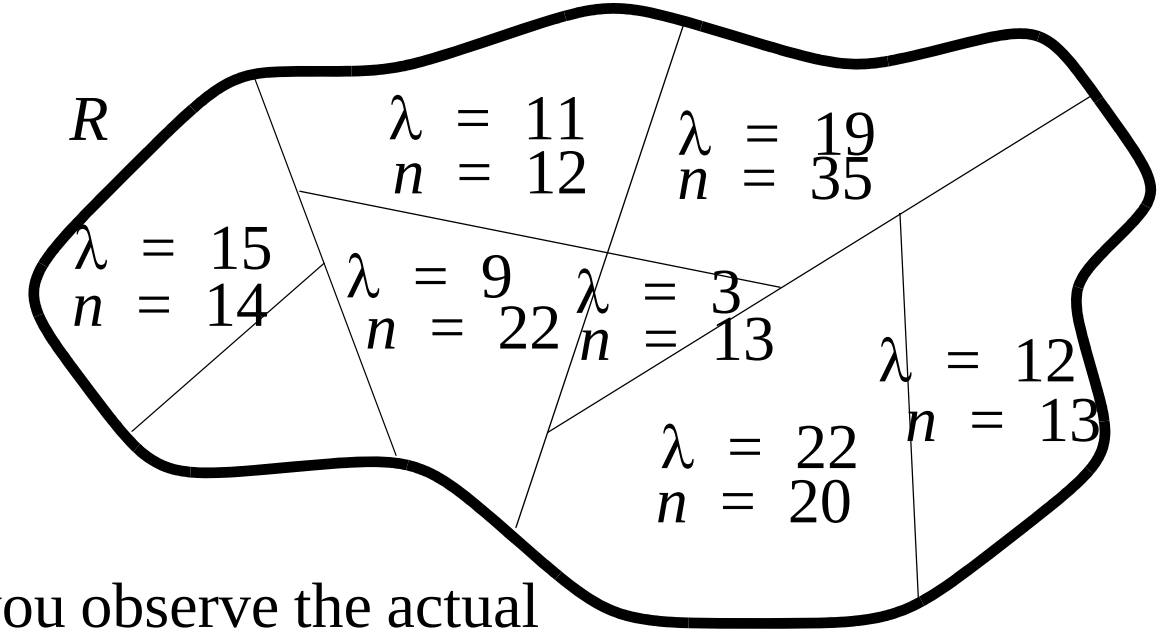
- Underlying statistical model is typically quite simple



Assume data in each cell is generated using simple dist (Poisson) with known params

Spatial Anomaly Detection

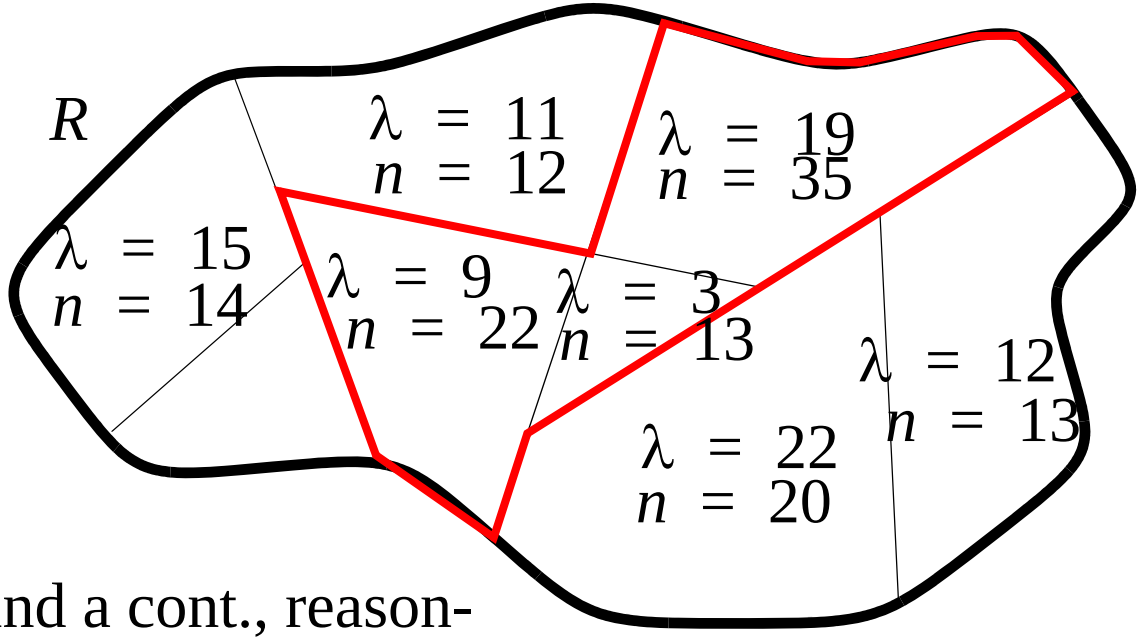
- Underlying statistical model is typically quite simple



Then when you observe the actual data...

Spatial Anomaly Detection

- Underlying statistical model is typically quite simple



...you try to find a cont., reasonably-shaped region where obs. data are extremely unlikely

Spatial Anomaly Detection

- Most related work in the mining literature begins with Kuldorff's SSS (Poisson model)...

  -Idea is to search all possible contiguous regions

  -Find top k that reject a Poisson-based LRT

  -Maybe to simulation to deal with MHT

  -Miners try to speed it using clever computational methods

  -ex: Neil et. al, Agarwal et. al, ...

More Complicated Models

- But what if your problem is more complicated?
- Our motivating example:

  -Anomalies in nosacomial antimicrobial resistance trends

  -Due to (mis-)use of antimocrobials, bugs develop resistance

  -But is the upward trend uniform, or is there spatial variation?

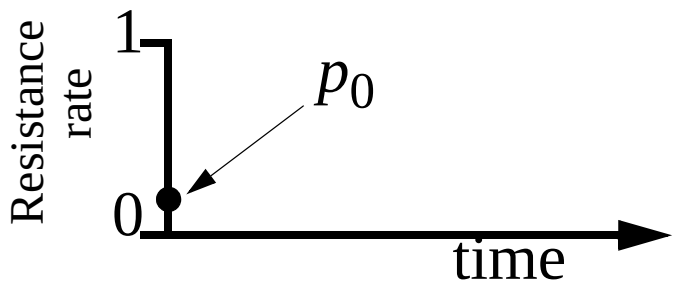
More Complicated Models

- But what if your problem is more complicated?
- Our motivating example:

  -Anomalies in nosacomial antimicrobial resistance trends

  -Due to (mis-)use of antimocrobials, bugs develop resistance

  -But is the upward trend uniform, or is there spatial variation?
- A reasonable model for a hospital's resistance trend:



At start of time, resistance probability for a bug in an arbitrary patient is p_0

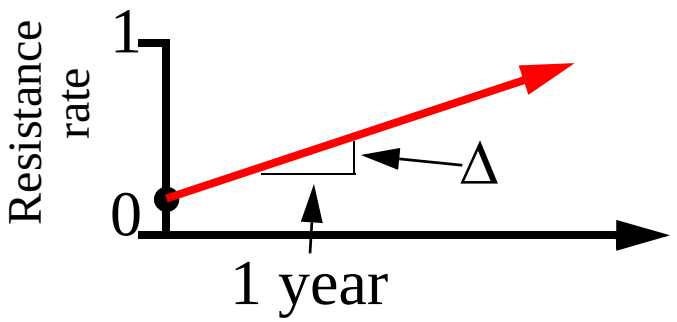
More Complicated Models

- But what if your problem is more complicated?
- Our motivating example:

  -Anomalies in nosacomial antimicrobial resistance trends

  -Due to (mis-)use of antimicrobials, bugs develop resistance

  -But is the upward trend uniform, or is there spatial variation?
- A reasonable model for a hospital's resistance trend:



Each year, there is a change Δ in resistance rate

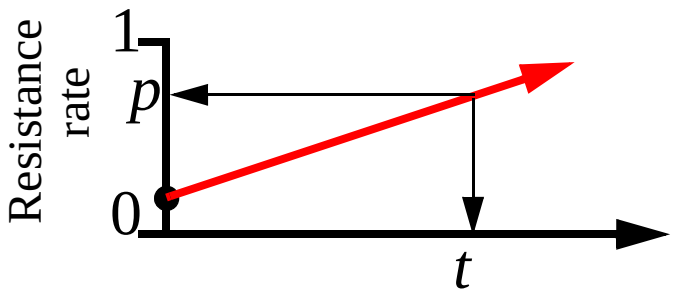
More Complicated Models

- But what if your problem is more complicated?
- Our motivating example:

  -Anomalies in nosacomial antimicrobial resistance trends

  -Due to (mis-)use of antimicrobials, bugs develop resistance

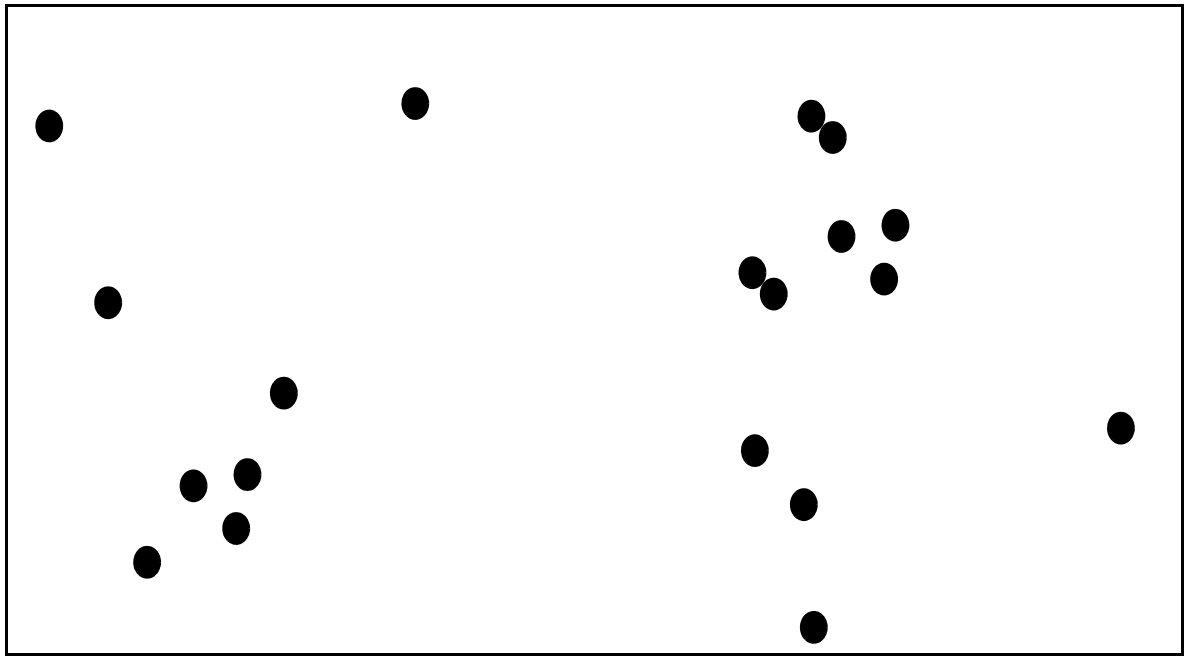
  -But is the upward trend uniform, or is there spatial variation?
- A reasonable model for a hospital's resistance trend:



An infected patient at time t has a resistant bug if Bernoulli trial with prob $p = p_0 + (t - t_0)\Delta$ is true

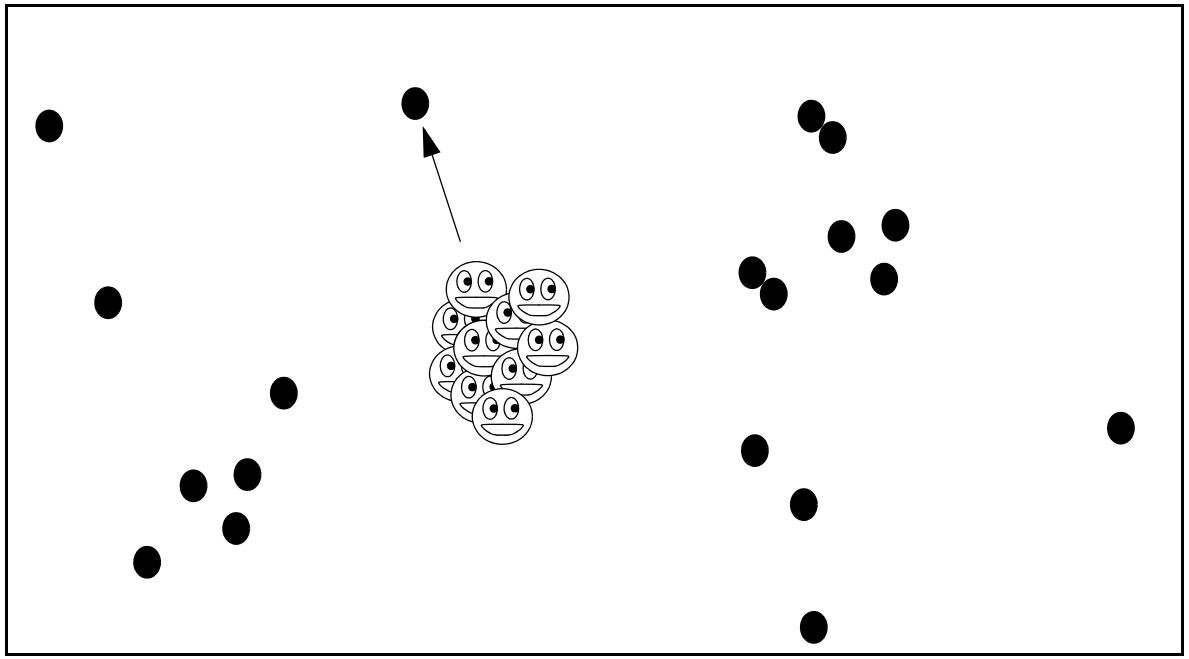
Finding a Region of Unique Trends

- Given this, we have many hospitals in a large spatial area



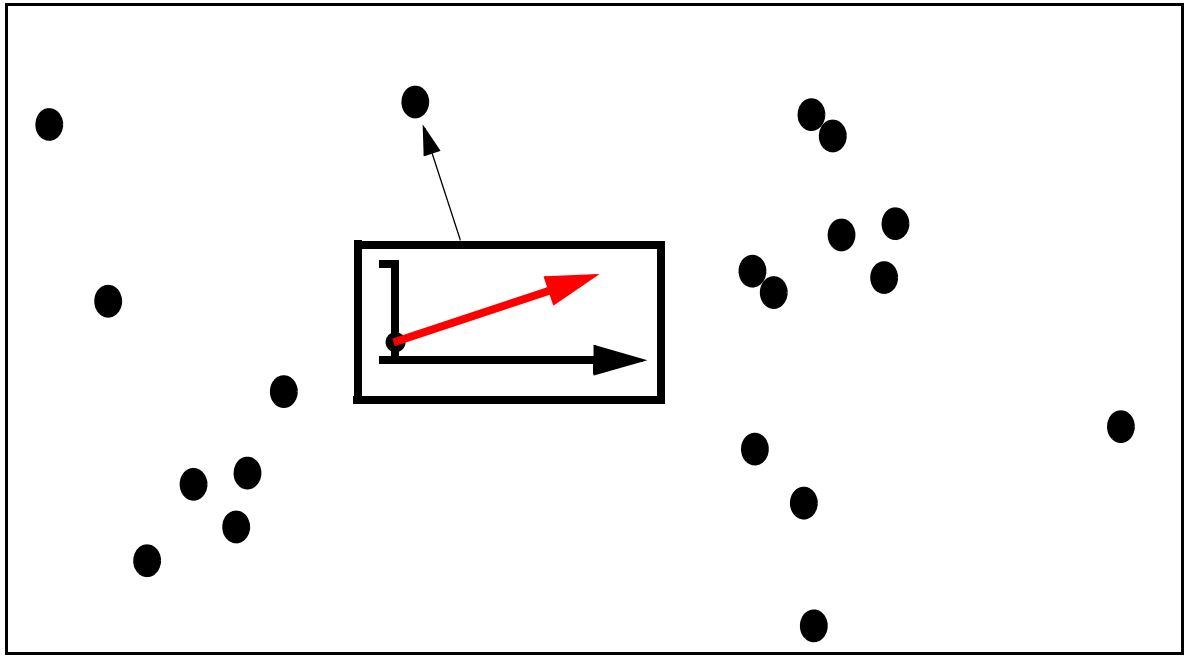
Finding a Region of Unique Trends

- Each has its own set of data associated with it



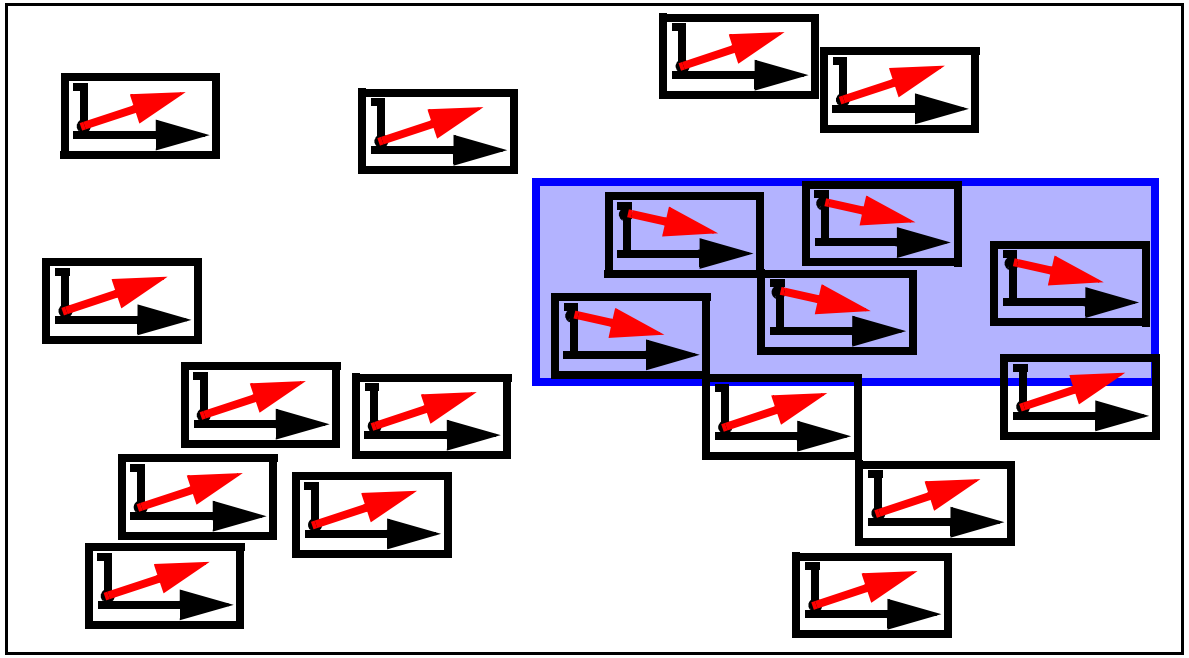
Finding a Region of Unique Trends

- Data are used to learn a local value for Δ



Finding a Region of Unique Trends

- Then we find any local region with an abnormal Δ



Two Key Problems to Address

- This is just one application where a reasonably rich, application-specific model is useful for anomaly search
- We are convinced that there are many apps that would benefit from the ability to apply their own models
- Gives rise to two key questions:

1. How does one build a generic software that allows spatial anomaly search with virtually any user-specified statistical model?

   -Is there a principled way to define the general detection problem?

   -How can a user specify his/her specific model?

2. How does one ensure that the search is reasonably fast?

Building a General Purpose Software - Problem Definition

- Actually quite easy to come up with an appropriate problem definition, based on a generic LRT (SSS is an LRT, too)
- LRT:

  -Given likelihood function $L(\theta|X)$

  -Let Θ be the full parameter space, Θ_0 the "null" or uninteresting part of the parameter space (e.g., all Δs are identical)

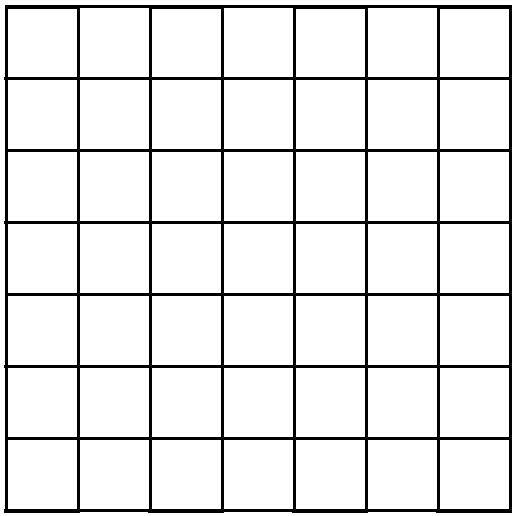
  -Want to compare $H_0: \theta \in \Theta_0$ vs. $H_a: \theta \in \Theta - \Theta_0$

  -Can use $\Lambda(X) = -2\log \dfrac{\sup_{\theta \in \Theta_0} L(\theta|X)}{\sup_{\theta \in \Theta} L(\theta|X)}$

  -Classic result; under H_0, $\Lambda(X)$ is chi-squared

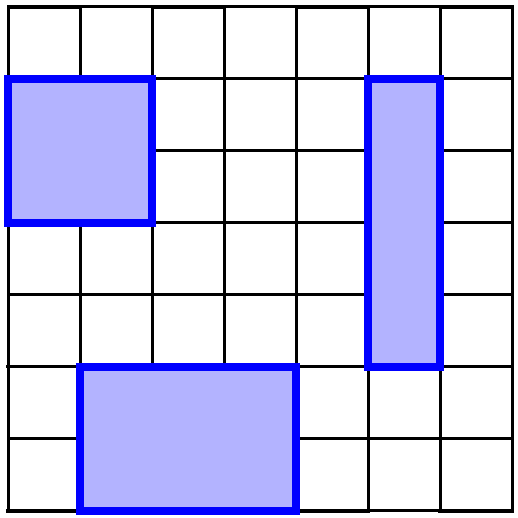
Building a General Purpose Software - Problem Definition

- Then, lay out data in a grid:



Building a General Purpose Software - Problem Definition

- Then, lay out data in a grid:



$k = 3$

- And find the k rectangular regions with the greatest $\Lambda(X)$ values
- Can use chi-squared dist to determine significance, or else do simulation

“Templatizing” the Software

- That’s fine, but an MS in stats could come up with this, not much of an interesting contribution!
- What we want to do is to build a software that makes it easy to implement such a test for any particular likelihood function

“Templatizing” the Software

- To apply our software, user must first develop an appropriate generative statistical model (binomial with time-dependent *p* in our example) - only restriction is independence across cells
- Model parameters must then be categorized as follows:

  -“Shared parameters”: params fixed across all cells by Θ_0

  -“Test parameters”: subset of shared params that are indicative of an anomalous region (they *can* vary in Θ) (e.g., Δ)

  -“Local parameters”: params allowed to vary by cell

  -Locals may be known (e.g., number of patients) or unknown and must be inferred (starting resistance rate)

- Then we seek to find a region *A* that strongly rejects the hypothesis that the test parameters are uniform across the boundary of *A*

“Templatizing” the Software

- User must then supply four template functions:

(1) The summarizing function $f(A)$ - accepts a region A and returns the summary data X_A associated with A, as well as and known constant-valued local params

“Templatizing” the Software

- User must then supply four template functions:

(1) The summarizing function $f(A)$ - accepts a region A and returns the summary data X_A associated with A, as well as and known constant-valued local params

(2) The likelihood function $L(\theta|X)$

“Templatizing” the Software

- User must then supply four template functions:

(1) The summarizing function $f(A)$ - accepts a region A and returns the summary data X_A associated with A, as well as and known constant-valued local params

(2) The likelihood function $L(\theta|X)$

(3) The null-space MLE procedure $MLE_0(f(A))$ - that chooses the set of parameter values from Θ_0 that maximize $L(\theta|X_A)$

“Templatizing” the Software

- User must then supply four template functions:

(1) The summarizing function $f(A)$ - accepts a region A and returns the summary data X_A associated with A, as well as and known constant-valued local params

(2) The likelihood function $L(\theta|X)$

(3) The null-space MLE procedure $MLE_0(f(A))$ - that chooses the set of parameter values from Θ_0 that maximize $L(\theta|X_A)$

(4) The complete-space MLE procedure $MLE_1(f(A))$ that chooses its parameter values from Θ - the params in the test set can vary across the boundary of A

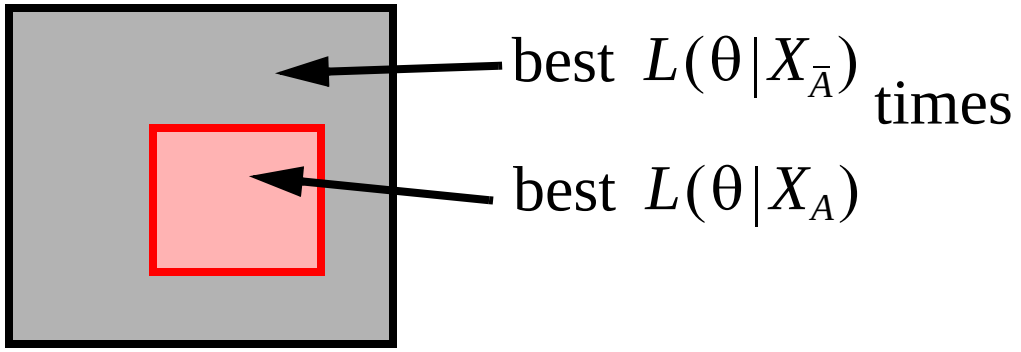
- Then the software does the rest!

Key Implementation Challenge - Speed

- There are ~n^4 rectangles to compute $\Lambda(X_A)$ for
- What's a realistic but relatively large value of n? Maybe 100?
- Granted, when I've got Amazon's cloud, 100^4 is not that big
- But naive implementation invokes MLE_0, MLE_1 for each A
- MLEs often need non-linear optimization; at ten seconds each, that's still 32 years of compute time - big even by cloud standards

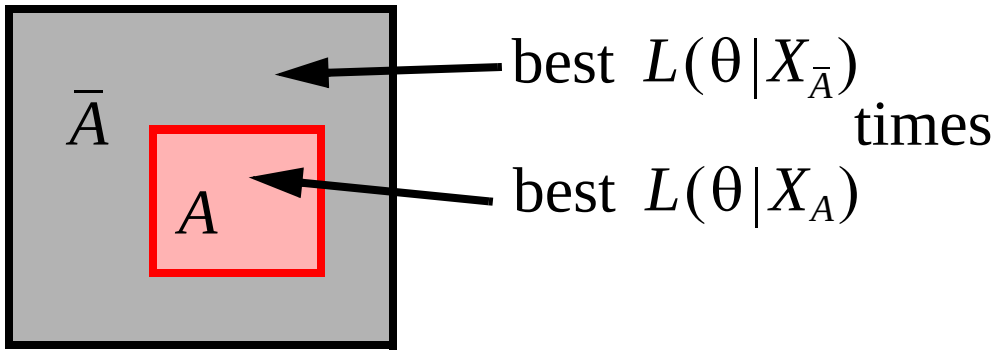
Achieving Speed

- We need to compute $\Lambda(X_A) = -2\log \frac{\sup_{\theta \in \Theta_0} L(\theta|X_A)}{\sup_{\theta \in \Theta} L(\theta|X_A)}$
- Numerator is easy: same for all *A*
- Can upper bound denominator... how?
- For a given *A*, we want:

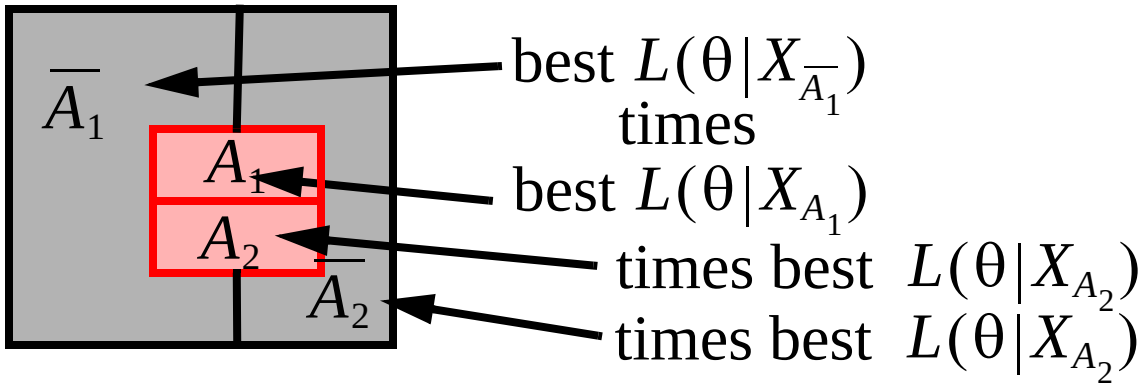


Achieving Speed

- Intuitively: add more parameters, you increase the supremum...
- So we know that:

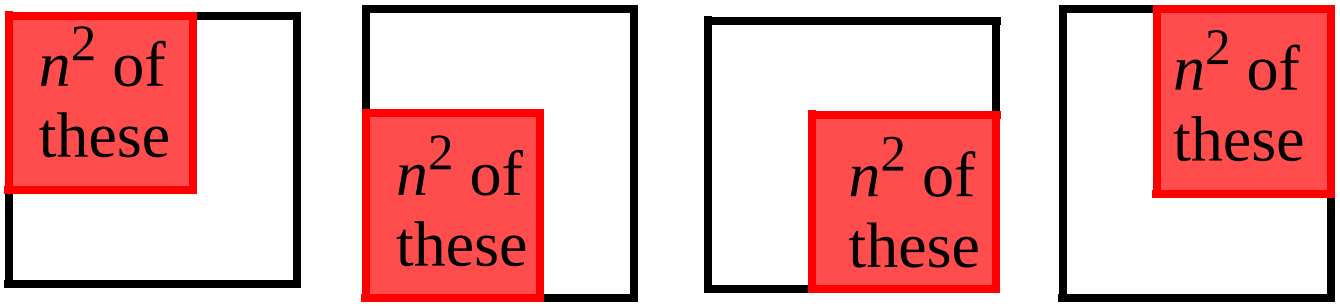


can't be as good as:

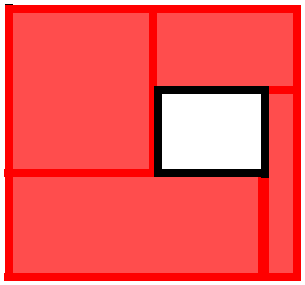


Achieving Speed

- So we carefully choose a number of rectangles and precompute their supremums; during search, can prune vast majority of rects
- Example: precompute all $4n^2$ rects of the form:



- Then can tile any $\bar{A}$ to upper bound MLE_1 .



with perfect pruning, O(n^2) reduction in optimization calls...

How Well Does it Work?

- We have applied this to many problems; a few of which are described in the paper
- In general, speedup is in the range of 5 times to 100 times
- In practice, two sources of speedup

  -High pruning rate + O(n^3) precomp = fast

  -Precomp tends to be over smaller rects, so faster MLE
- On the antimicrobial resistance example (300+ hospitals):

Grid size	Our time	Pruning rate	Naive time	Speedup
16 x 16	0.15 days	96.5%	2.6 days	17.3
32 x 32	1.1 days	97.6%	36 days	31.8
64 x 64	11.9 days	98.0%	544 days	45.7

Conclusions

- If you have your own spatial anomaly detection algorithm, it’s easy to implement a fast solution using our software
- We won’t beat a special-purpose software coded for a Poisson model... (are those really useful, anyway? --Amazon!)
- ...but for complicated models our pruning can dramatically reduce the search time
- Plus it employs a principled, time-tested approach to parameter comparison: the LRT
- Questions?