Fast Subset Scanning for Scalable Event and Pattern Detection

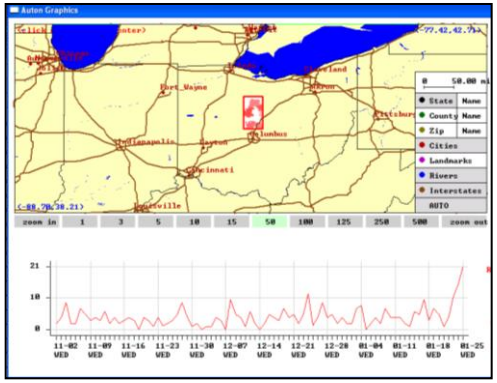
Daniel B. Neill
H.J. Heinz III College
Carnegie Mellon University
E-mail: neill@cs.cmu.edu

We gratefully acknowledge funding support from the National Science Foundation, grants IIS-0916345, IIS-0911032, and IIS-0953330.

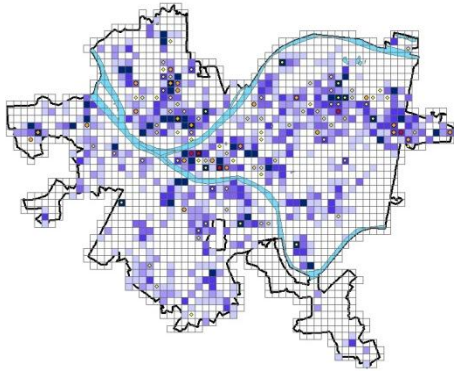
Daniel B. Neill (neill@cs.cmu.edu)
Associate Professor of Information Systems, Heinz College
Courtesy Associate Professor of Machine Learning and Robotics, SCS

My research has two main goals: to develop new machine learning methods for automatic **detection** of **events** and other **patterns** in massive datasets, and to apply these methods to improve the quality of public health, safety, and security.

<u>Customs monitoring</u>: detecting patterns of illicit container shipments



<u>Biosurveillance</u>: early detection of emerging outbreaks of disease



<u>Law enforcement</u>: detection and prediction of crime hot-spots

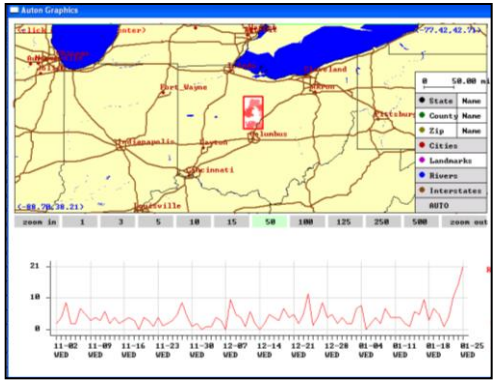
Our methods could have detected the May 2000 Walkerton *E. coli* outbreak two days earlier than the first public health response.

We are able to accurately predict emerging clusters of violent crime, a week in advance, by detecting clusters of more minor “leading indicator” crimes.

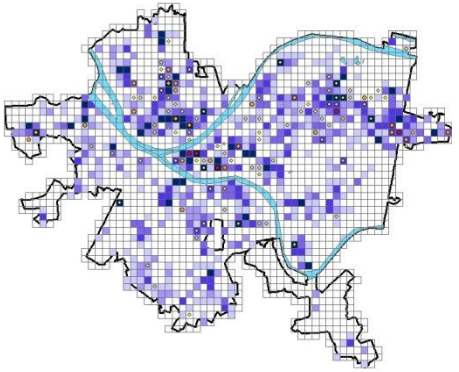
Daniel B. Neill (neill@cs.cmu.edu)
Associate Professor of Information Systems, Heinz College
Courtesy Associate Professor of Machine Learning and Robotics, SCS

My research has two main goals: to develop new machine learning methods for automatic **detection** of **events** and other **patterns** in massive datasets, and to apply these methods to improve the quality of public health, safety, and security.

<u>Customs monitoring</u>: detecting patterns of illicit container shipments



<u>Biosurveillance</u>: early detection of emerging outbreaks of disease



<u>Law enforcement</u>: detection and prediction of crime hot-spots

Our methods are currently in use for deployed biosurveillance systems in Ottawa and Grey-Bruce, Ontario; several other projects are underway.

We collaborate directly with the Chicago Police Department, and our “CrimeScan” software is in successful, day-to-day operational use for predictive policing.

Pattern detection by subset scan

One key insight that underlies much of my work is that pattern detection can be viewed as a **search** over subsets of the data.

Statistical challenges:
Which subsets to search?
Is a given subset anomalous?
Which anomalies are relevant?

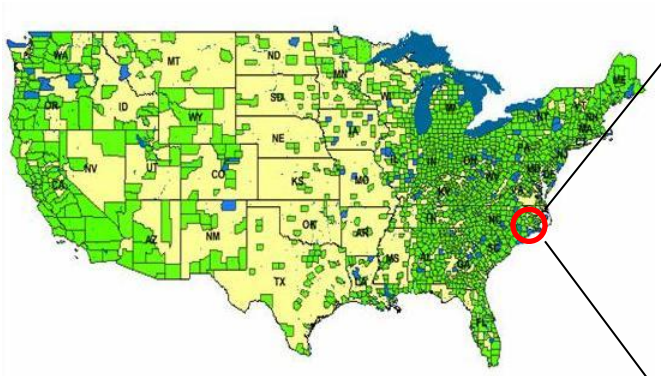
Computational challenge:
How to make this search over subsets efficient for massive, complex, high-dimensional data?

New statistical methods enable more timely and more accurate detection by integrating **multiple data sources**, incorporating **spatial** and **temporal** information, and using **prior knowledge** of a domain.

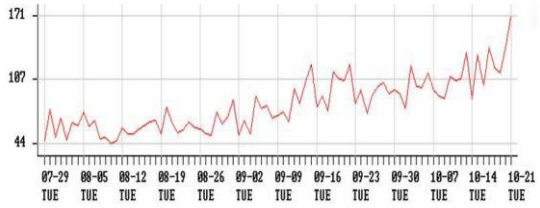
New algorithms and data structures make previously impossible detection tasks computationally feasible and fast.

New machine learning methods enable our systems to learn from user feedback, modeling and distinguishing between relevant and irrelevant types of anomaly.

Multivariate event detection



Spatial time series data from spatial locations s_i (e.g. zip codes)



Time series of counts $c_{i,m}{}^t$ for each zip code s_i for each data stream d_m.

Outbreak detection

d_1 = respiratory ED

d_2 = constitutional ED

d_3 = OTC cough/cold

d_4 = OTC anti-fever

(etc.)

Main goals:

Detect any emerging events.

Pinpoint the affected subset of locations and time duration.

Characterize the event by identifying the affected streams.

Compare hypotheses:

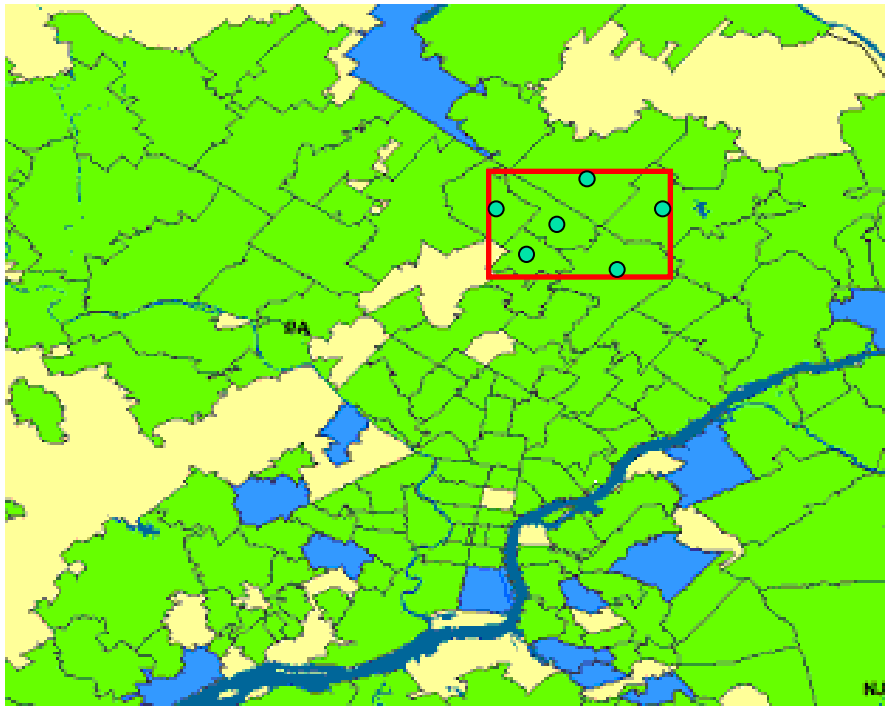
$H_1(D, S, W)$

D = subset of streams
S = subset of locations
W = time duration

vs. H_0: no events occurring

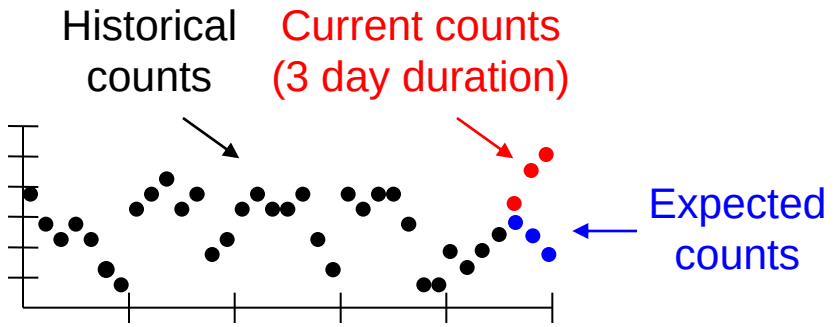
Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)



We search for spatial regions (subsets of locations) where the recently observed counts for some subset of streams are significantly higher than expected.

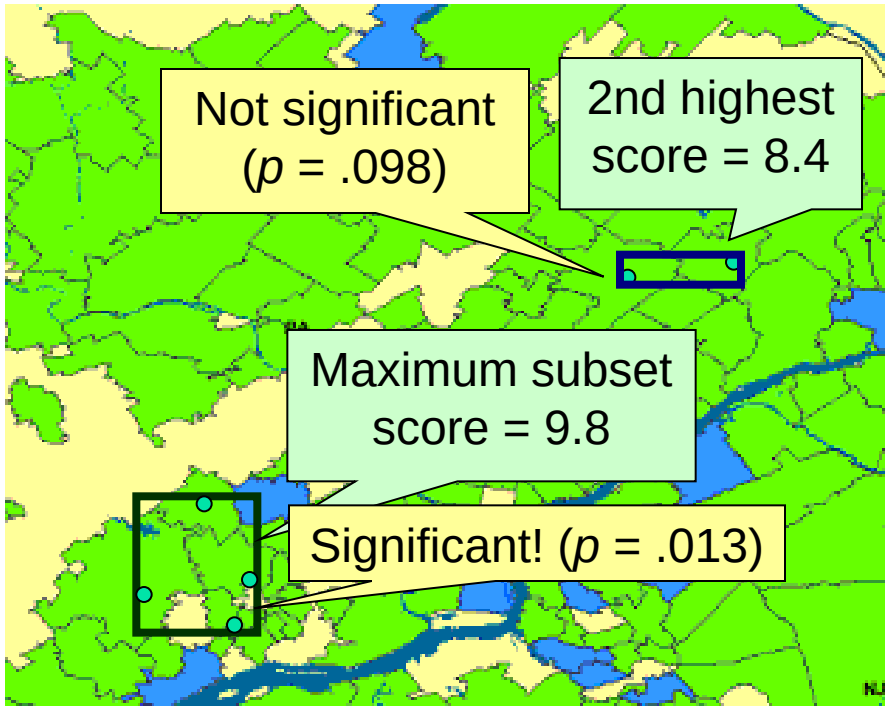
We perform **time series analysis** to compute expected counts ("baselines") for each location and stream for each recent day.



We then compare the actual and expected counts for each subset (D, S, W) under consideration.

Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

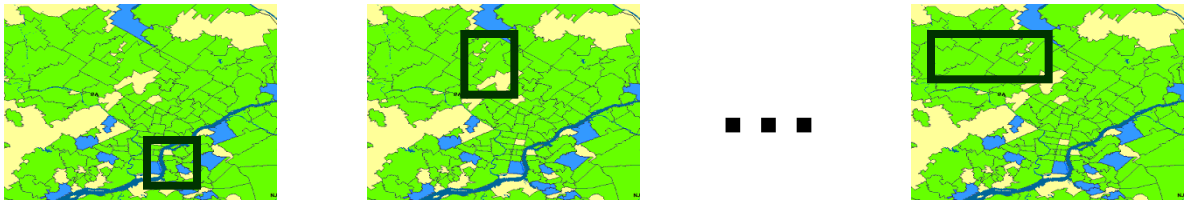


We find the subsets with highest values of a likelihood ratio statistic, and compute the *p*-value of each subset by randomization testing.

$$\mathrm{F}(D,S,W) = \frac{\Pr(\mathrm{Data} \mid H_1(D,S,W))}{\Pr(\mathrm{Data} \mid H_0)}$$

<u>To compute p-value</u>
Compare subset score to maximum subset scores of simulated datasets under H_0.

$F_1^* = 2.4$ $F_2^* = 9.1$ ... $F_{999}^* = 7.0$



Likelihood ratio statistics

For our expectation-based scan statistics, the null hypothesis H_0 assumes "business as usual": each count $c_{i,m}^t$ is drawn from some parametric distribution with mean $b_{i,m}^t$. $H_1(S)$ assumes a multiplicative increase for the affected subset S.

<u>Expectation-based Poisson</u>

H_0: $c_{i,m}^t \sim \text{Poisson}(b_{i,m}^t)$

$H_1(S)$: $c_{i,m}^t \sim \text{Poisson}(qb_{i,m}^t)$

Let $C = \sum_S c_{i,m}^t$ and $B = \sum_S b_{i,m}^t$.

Maximum likelihood: $q = C / B$.

$F(S) = C \log (C/B) + B - C$

<u>Expectation-based Gaussian</u>

H_0: $c_{i,m}^t \sim \text{Gaussian}(b_{i,m}^t, \sigma_{i,m}^t)$

$H_1(S)$: $c_{i,m}^t \sim \text{Gaussian}(qb_{i,m}^t, \sigma_{i,m}^t)$

Let $C' = \sum_S c_{i,m}^t b_{i,m}^t / (\sigma_{i,m}^t)^2$ and $B' = \sum_S (b_{i,m}^t)^2 / (\sigma_{i,m}^t)^2$.

Maximum likelihood: $q = C' / B'$.

$F(S) = (C')^2 / 2B' + B'/2 - C'$

Many more possibilities: any single parameter exponential family, or nonparametric.

Which regions to search?

- Typical approach: each search region S is a subregion of the search space.
  - Choose some region shape (e.g. circles, rectangles) and consider all regions of that shape and varying size.
  - Low power for true events that do not correspond well to the chosen set of search regions (e.g. irregular shapes).
- Alternate approach: each search region S represents a distinct subset of the N locations.
  - Find the highest scoring subset, subject to some constraints (e.g. spatial proximity, connectivity).
  - For multivariate, also optimize over subsets of the M monitored data streams.
  - Exponentially many possible subsets, $O(2^N \times 2^M)$: computationally infeasible for naïve search.

Fast subset scan

- In certain cases, we can optimize F(S) over the exponentially many subsets of locations, while evaluating only O(N) rather than $O(2^N)$ subsets.
- Many commonly used scan statistics have the property of linear-time subset scanning:
  - Just sort the locations from highest to lowest priority according to some function…
  - … then search over groups consisting of the top-k highest priority locations, for k = 1..N.

The highest scoring subset is guaranteed to be one of these!

Sample result: we can find the **most anomalous** subset of Allegheny County zip codes in 0.03 sec vs. 10^{24} years.

Linear-time subset scanning

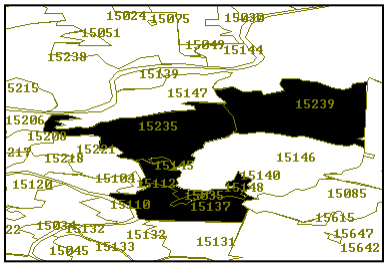
- Example: Expectation-Based Poisson statistic
  - Sort data locations s_i by the ratio of observed to expected count, c_i / b_i.
  - Given the ordering $s_{(1)} \ldots s_{(N)}$, we can **prove** that the top-scoring subset F(S) consists of the locations $s_{(1)} \ldots s_{(k)}$ for some k, $1 \leq k \leq N$.
  - Proof by "inclusion": if there exists some location $s_{out} \notin S$ with higher priority than some location $s_{in} \in S$, then: $F(S) \leq \max(F(S \cup \{s_{out}\}), F(S \setminus \{s_{in}\}))$.
- Theorem: LTSS holds for convex functions of two additive sufficient statistics.
- Theorem: LTSS holds for all expectation-based scan statistics in any separable exponential family.

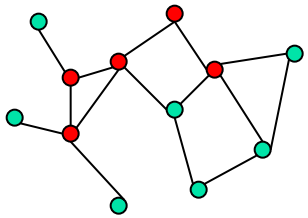
Fast Subset Scan for Pattern Detection

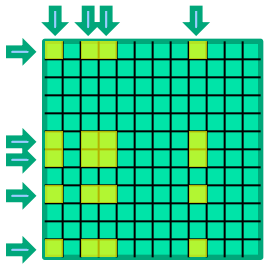
LTSS is a new and powerful tool for **exact** combinatorial optimization (as opposed to approximate techniques such as submodular function optimization). But it only solves the "best unconstrained subset" problem, and cannot be used directly for constrained optimization.

We are currently investigating how LTSS can be extended to the many real-world problems with (hard or soft) constraints on our search.

Proximity constraints	→	Fast spatial scan (irregular regions)
Multiple data streams	→	Fast multivariate scan
Connectivity constraints	→	Fast graph scan
Group self-similarity	→	Fast generalized subset scan







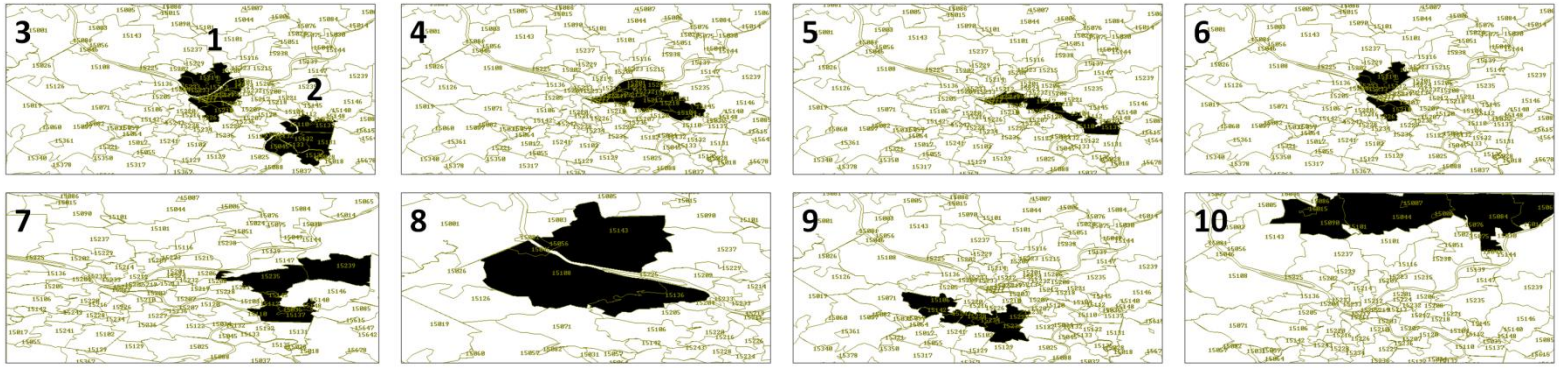
Other constraints? Shape, convexity, temporal consistency…
Other data types? Text, tensor data, dynamic graphs, etc.

Incorporating spatial proximity

- Maximize the spatial scan statistic over all subsets of the "local neighborhoods" consisting of a center location s_i and its k – 1 nearest neighbors, for a fixed neighborhood size k.
- Naïve search requires $O(N \cdot 2^k)$ time and is computationally infeasible for k > 25.
- For each center, we can search over all subsets of its local neighborhood in O(k) time using LTSS, thus requiring a total time complexity of O(Nk) + O(N log N) for sorting the locations.
- Variants: fixed radius r, **fast multiscan** (soft constraint: apply a linear penalty on k or r).

Evaluation on ED data

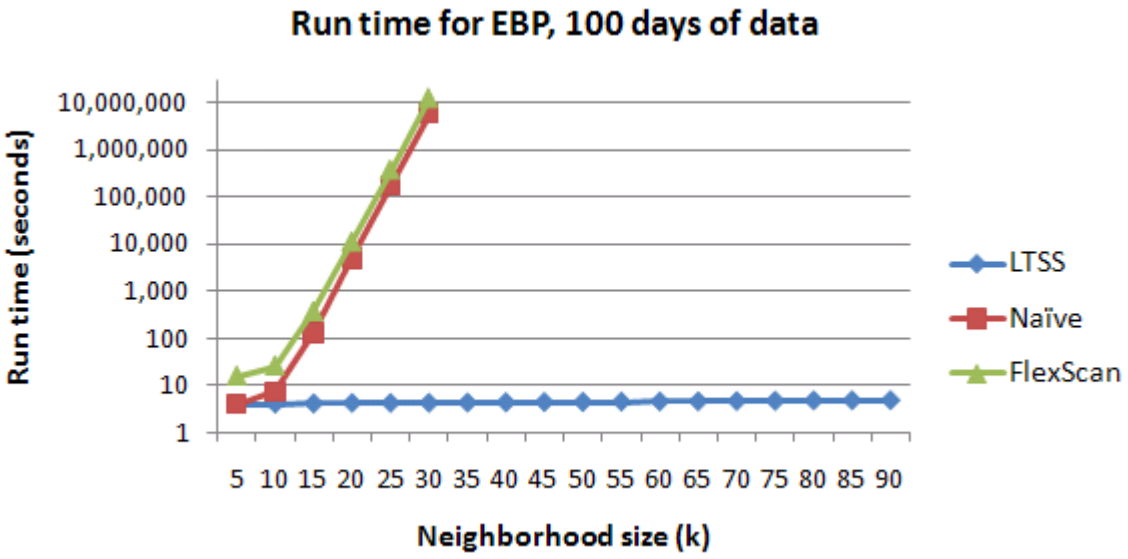
We injected simulated disease outbreaks of various shapes into real-world Emergency Department data.



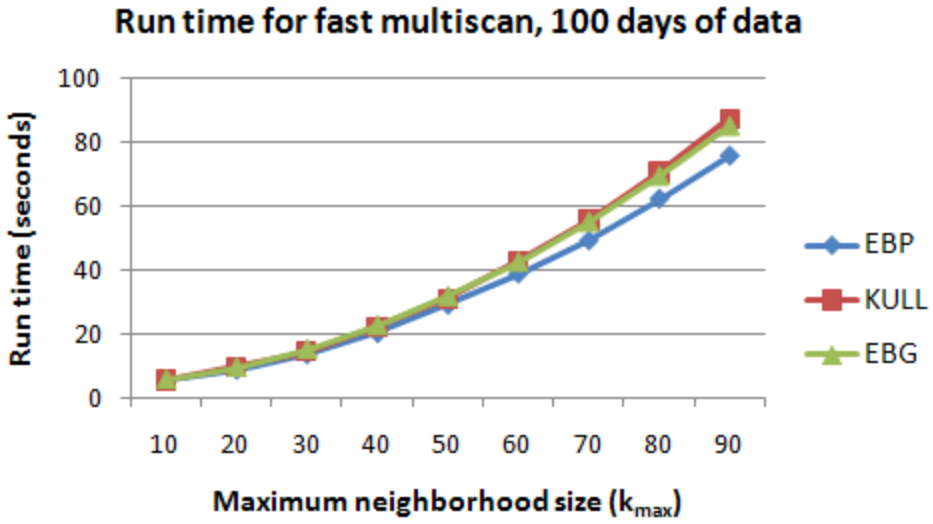
We compared the different methods in terms of:

- Run time
- Avg. time to outbreak detection vs. false positive rate
- Proportion of outbreaks detected vs. false positive rate
- Spatial accuracy (precision, recall, and overlap coeff.)

Comparison of run time

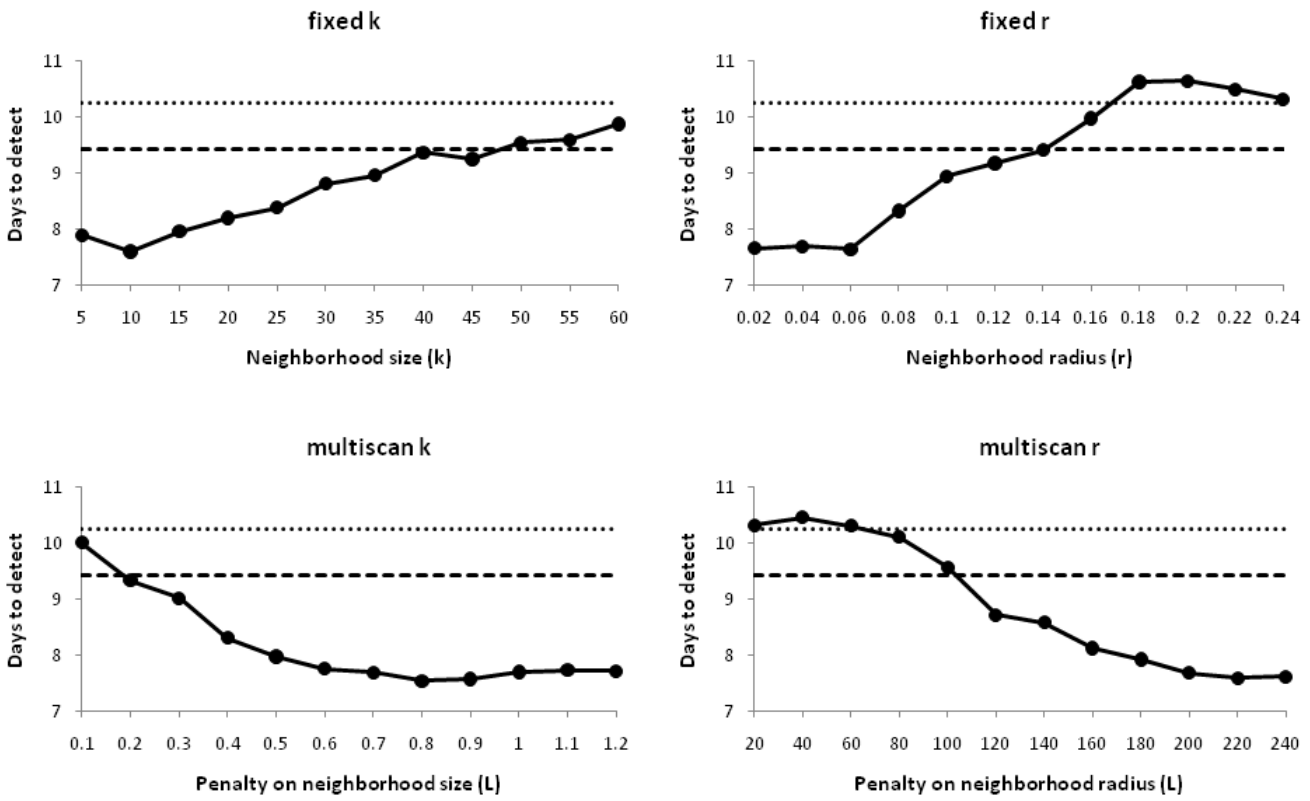


LTSS requires <50ms per day of data, vs. millions of years!



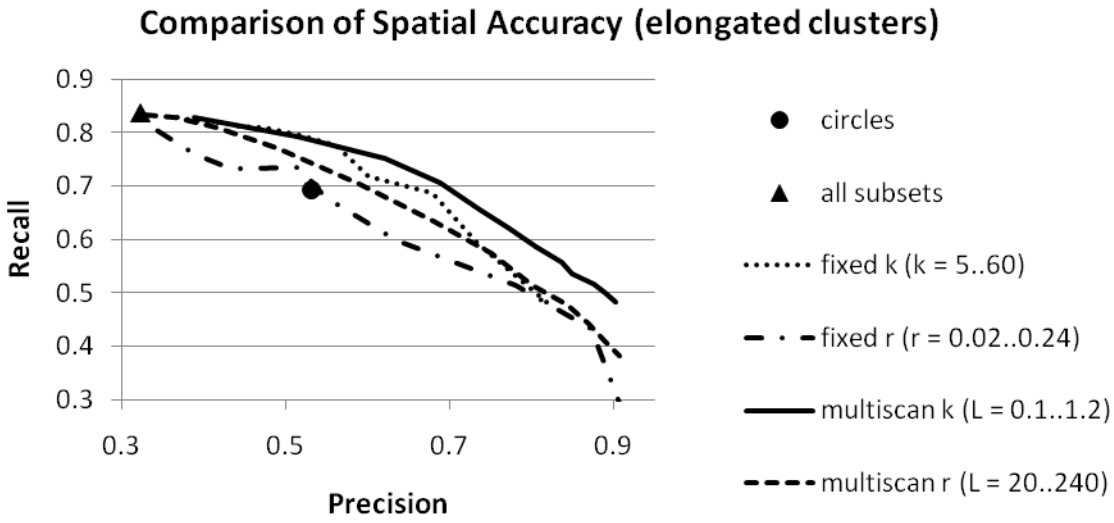
Fast multiscan requires less than one second per day of data.

Comparison of detection power



Average days to detect at 1 false positive/month, vs. "circles" (dashed) and "all subsets" (dotted). Our methods detected nearly **two days faster** than the standard circular scan, with fewer than half as many missed outbreaks.

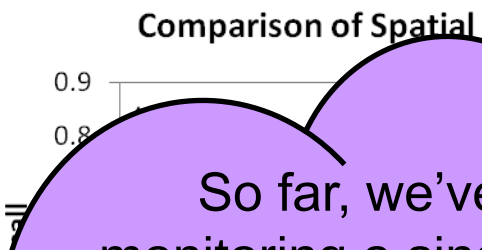
Comparison of spatial accuracy



For elongated or irregular clusters, our methods were able to achieve better spatial accuracy (higher precision, recall, and overlap coefficient) than the standard circular scan.

Not surprisingly, the circular scan had better spatial accuracy for clusters that were circular in shape.

Comparison of spatial accuracy



So far, we've focused on monitoring a single data stream.[1]

But we can also use LTSS to efficiently integrate information from many streams![2]

[1]Neill, *Journal of the Royal Statistical Society* (Series B), 2012.

[2]Neill, McFowland, and Zheng, *Statistics in Medicine*, 2013, in press.

For ble to call, a ar scan.

t st er spatial ccurac circular in shape.

Multivariate scan statistics

The univariate log-likelihood ratio statistic F(C, B) is a function of two aggregate sufficient statistics.

For the **expectation-based Poisson** (EBP) statistic:
$F(C, B) = C \log (C / B) + B - C$, if $C > B$, and 0 otherwise.

<u>Subset Aggregation multivariate spatial scan</u>

Assumes a **constant effect** over all affected data streams, computed by maximum likelihood estimation.

$$F(D, S, W) = F(\sum C_m, \sum B_m)$$

<u>Kulldorff's multivariate spatial scan</u>

Assumes **independent effects** on each data stream, each estimated separately by maximum likelihood.

$$F(D, S, W) = \sum F(C_m, B_m)$$

Sums are taken over all affected data streams $d_m \in D$.

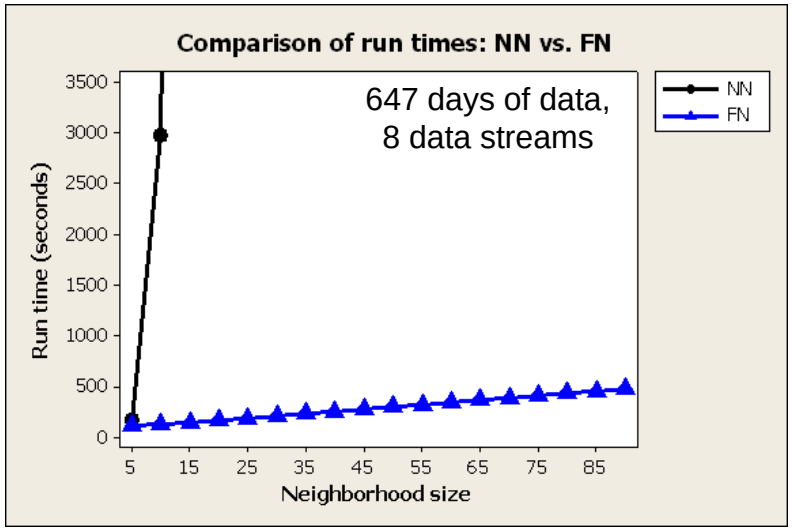
The count C_m and baseline B_m are aggregated over all affected spatial locations $s_i \in S$, for the given stream d_m and for the most recent W days of data.

Fast multivariate scans

How can we efficiently search over all subsets of data streams and over all proximity-constrained subsets of spatial locations?

Option 1 (fast/naïve, or FN): for each of the 2^M subsets of streams, aggregate counts and apply LTSS to efficiently search over subsets of locations.

Guaranteed to find the highest scoring subset!



For a fixed number of streams, FN fast localized scan scales linearly (not exponentially) with neighborhood size.

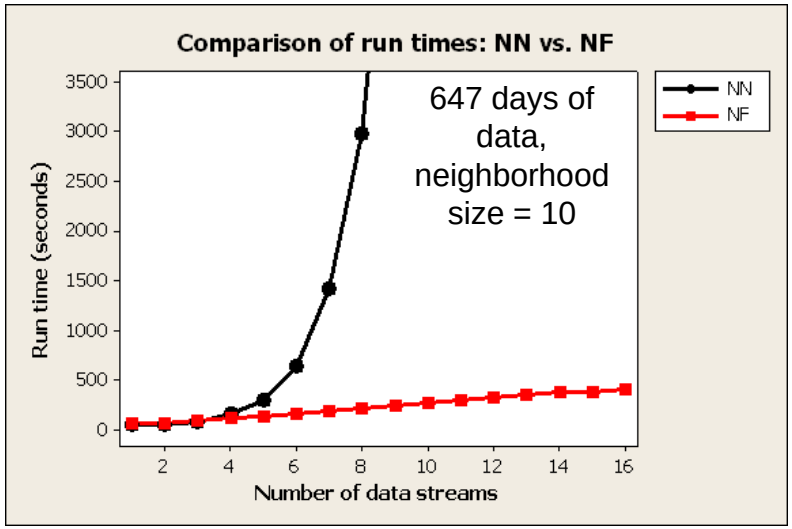
8 streams: <1 sec/day of data.

Fast multivariate scans

How can we efficiently search over all subsets of data streams and over all proximity-constrained subsets of spatial locations?

Option 2 (naïve/fast, or NF): exhaustively search over spatial regions. For each, perform efficient LTSS search over subsets of streams.

Guaranteed to find the highest scoring subset!



For a fixed neighborhood size k, NF fast localized scan scales linearly (not exponentially) with number of streams.

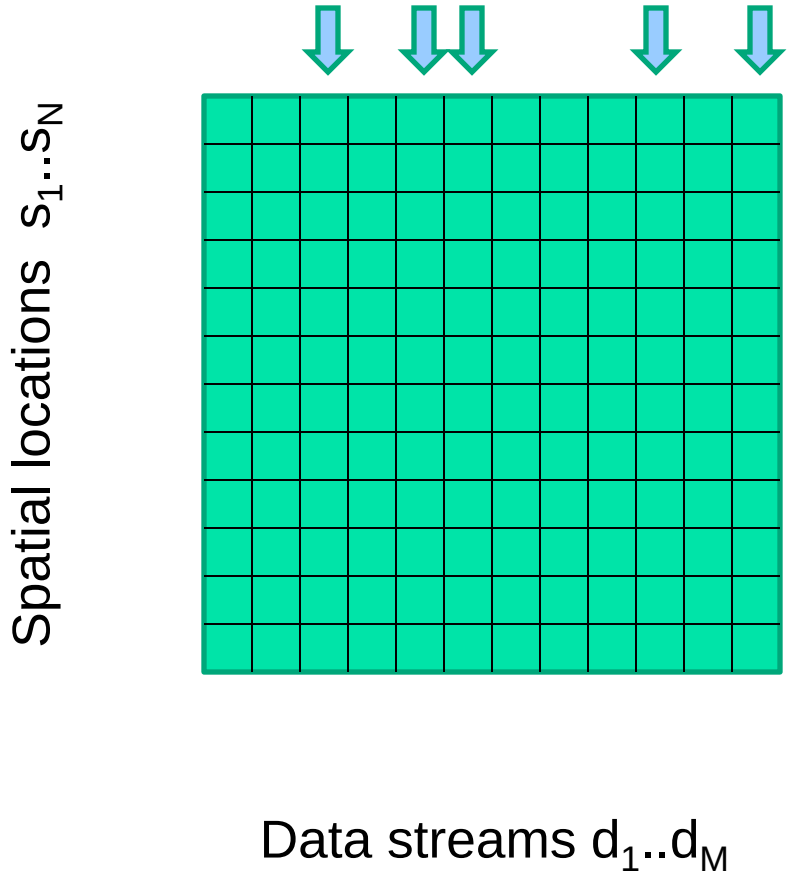
For k = 10: <1 sec/day of data

Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.



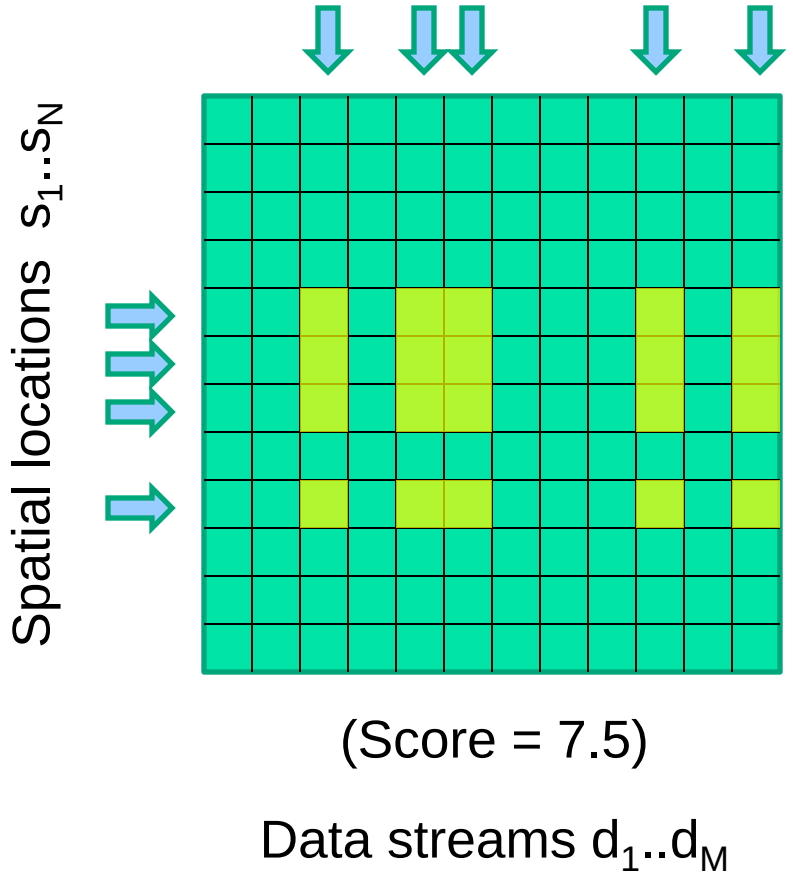
Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.



Fast multivariate scans

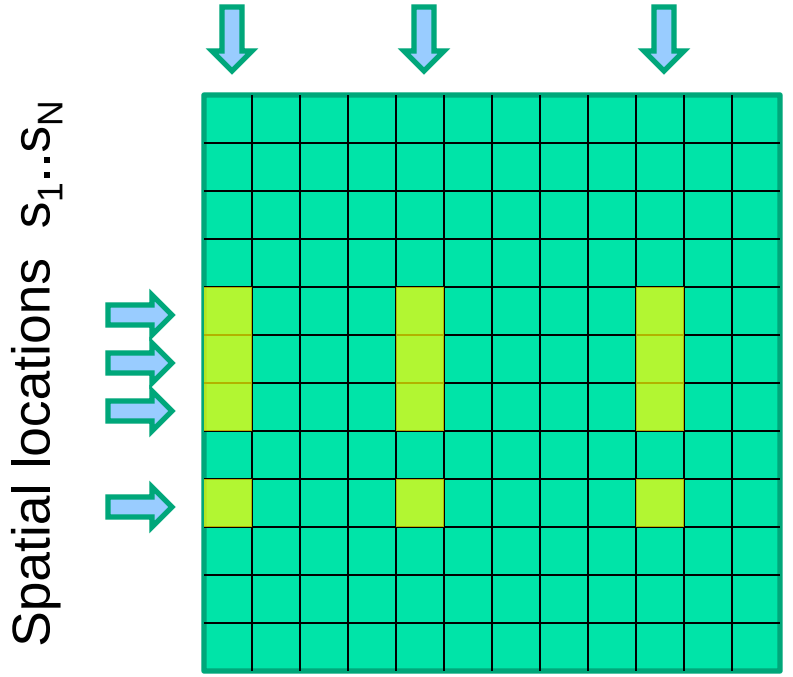
What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.



Fast multivariate scans

What if we have a large set of search regions and many data streams?

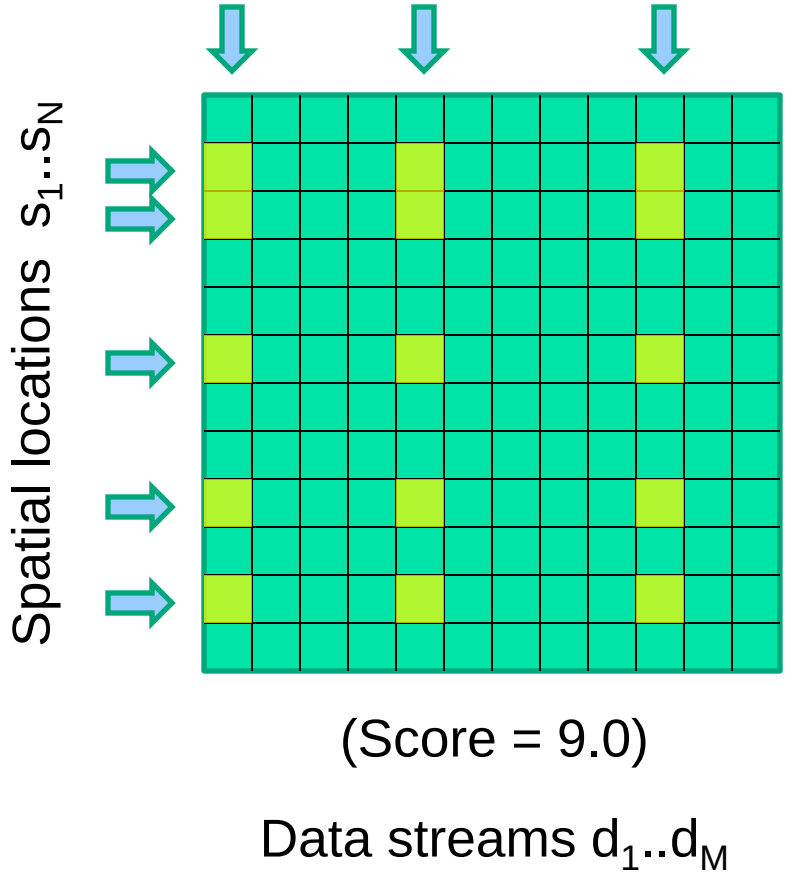
Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.



Fast multivariate scans

What if we have a large set of search regions and many data streams?

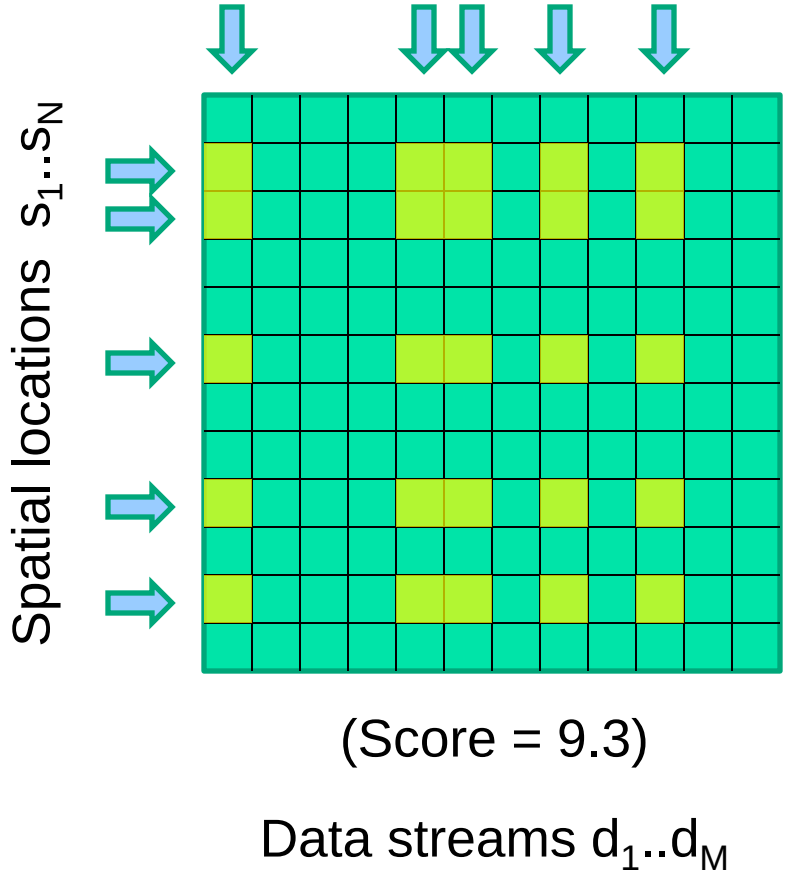
Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.



Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

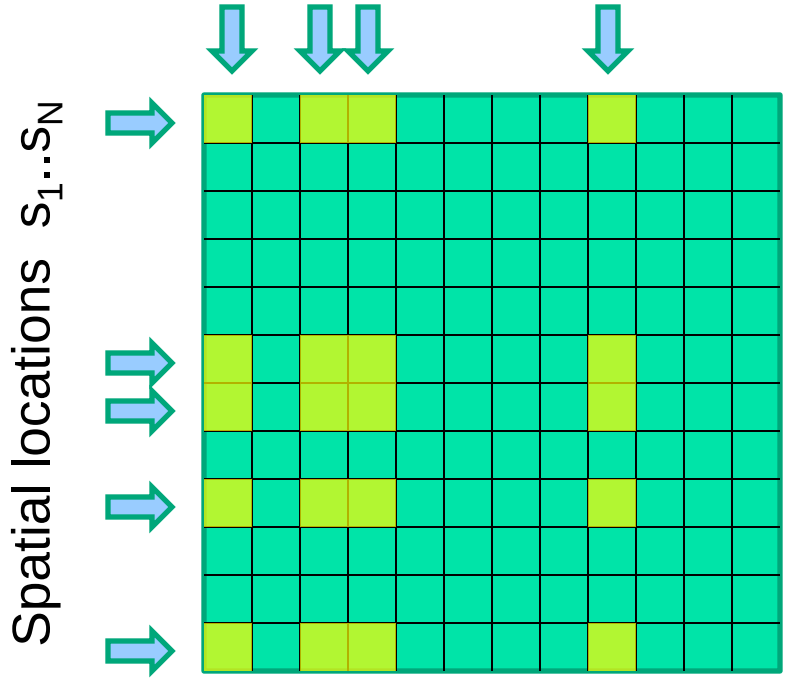
1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.

5. Repeat steps 1-4 for 50 random restarts.



Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

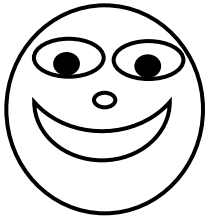
2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

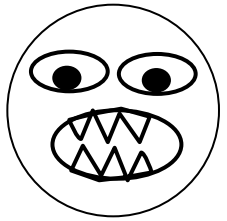
4. Iterate steps 2-3 until convergence.

5. Repeat steps 1-4 for 50 random restarts.

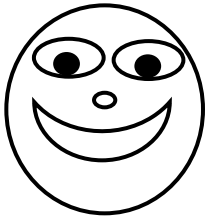
GOOD NEWS:
Run time is linear in number of locations & number of streams.



BAD NEWS:
Not guaranteed to find global maximum of the score function.



MORE GOOD NEWS:
200x faster than FN for 16 streams, and >98% approximation ratio.



SA vs. Kulldorff comparison

Using our new, fast algorithms, we evaluated the SA and Kulldorff multivariate scans on semi-synthetic outbreak detection tasks for 16 streams of Emergency Department data from Allegheny County, PA.

For both methods, searching over proximity-constrained subsets of locations resulted in 1 to 2 days faster detection, and significantly improved spatial accuracy (overlap), as compared to circular scan.

We observed an interesting tradeoff between the two methods' detection power and ability to characterize the affected streams.

Kulldorff's method tended to detect slightly faster than SA:
0.5 days for M = 2 streams, and 0.2 to 0.3 days for larger values of M.

But SA was better able to identify the affected subset of streams.

Discussion

The choice between the Subset Aggregation and Kulldorff versions of the multivariate spatial scan depends on whether our primary goal is **early detection** or **accurate characterization** of events.

Our fast algorithms, based on extensions of linear-time subset scanning to the multivariate case, enable either version to be computed efficiently, even for many locations and many streams.

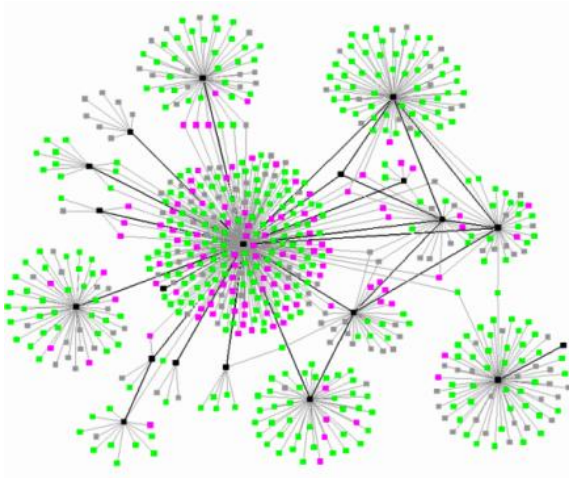
By scanning over all subsets of streams, and over all proximity-constrained subsets of locations, we can dramatically improve our ability to detect and characterize emerging outbreaks of disease.

For the Subset Aggregation scan, we have recently extended our FF algorithm to graph/network and tensor data, allowing us to scan over **connected** subsets of locations, **related** subsets of data streams, and **subpopulations** with different sets of demographic characteristics.

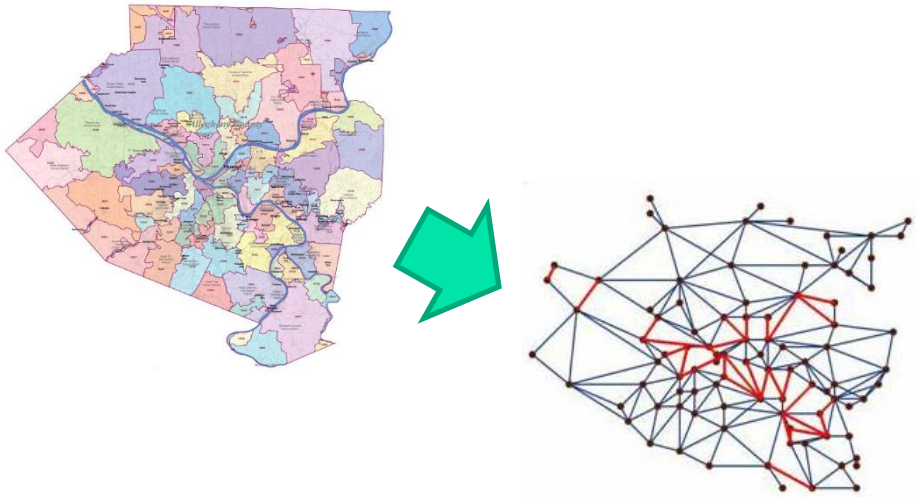
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data.

In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.



Example: tracking disease spread from person-to-person contact.



Example: identifying a **connected** subset of zip codes (Allegheny County, PA)

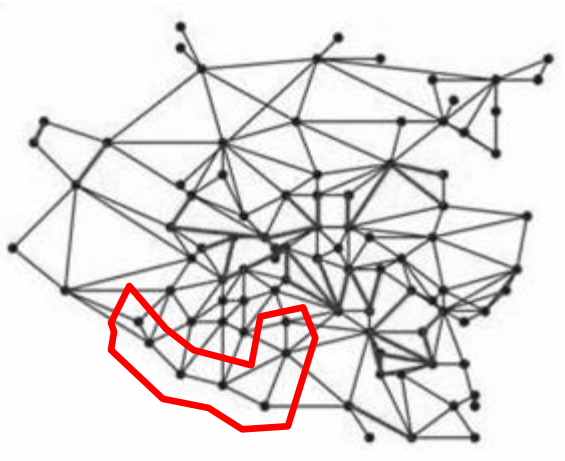
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data.

In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.

Our **GraphScan** algorithm* can efficiently and exactly identify the highest-scoring connected subgraph:

- Can incorporate multiple data streams
- With or without proximity constraints
- Graphs with several hundred nodes



We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

*Speakman and Neill, 2009; Speakman et al., 2013

Incorporating connectivity constraints

We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

LTSS property <u>without</u> connectivity constraints:
"If node $x \in S$ and node $y \notin S$, for $x > y$,
then subset S cannot be optimal."

We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Incorporating connectivity constraints

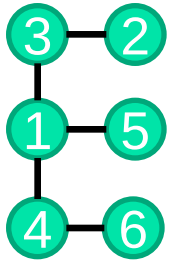
We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

LTSS property with connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
and S \ {x} and S U {y} are both connected,
then subset S cannot be optimal."



We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Incorporating connectivity constraints

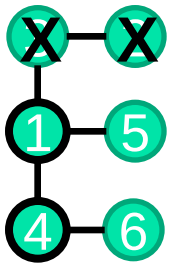
We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

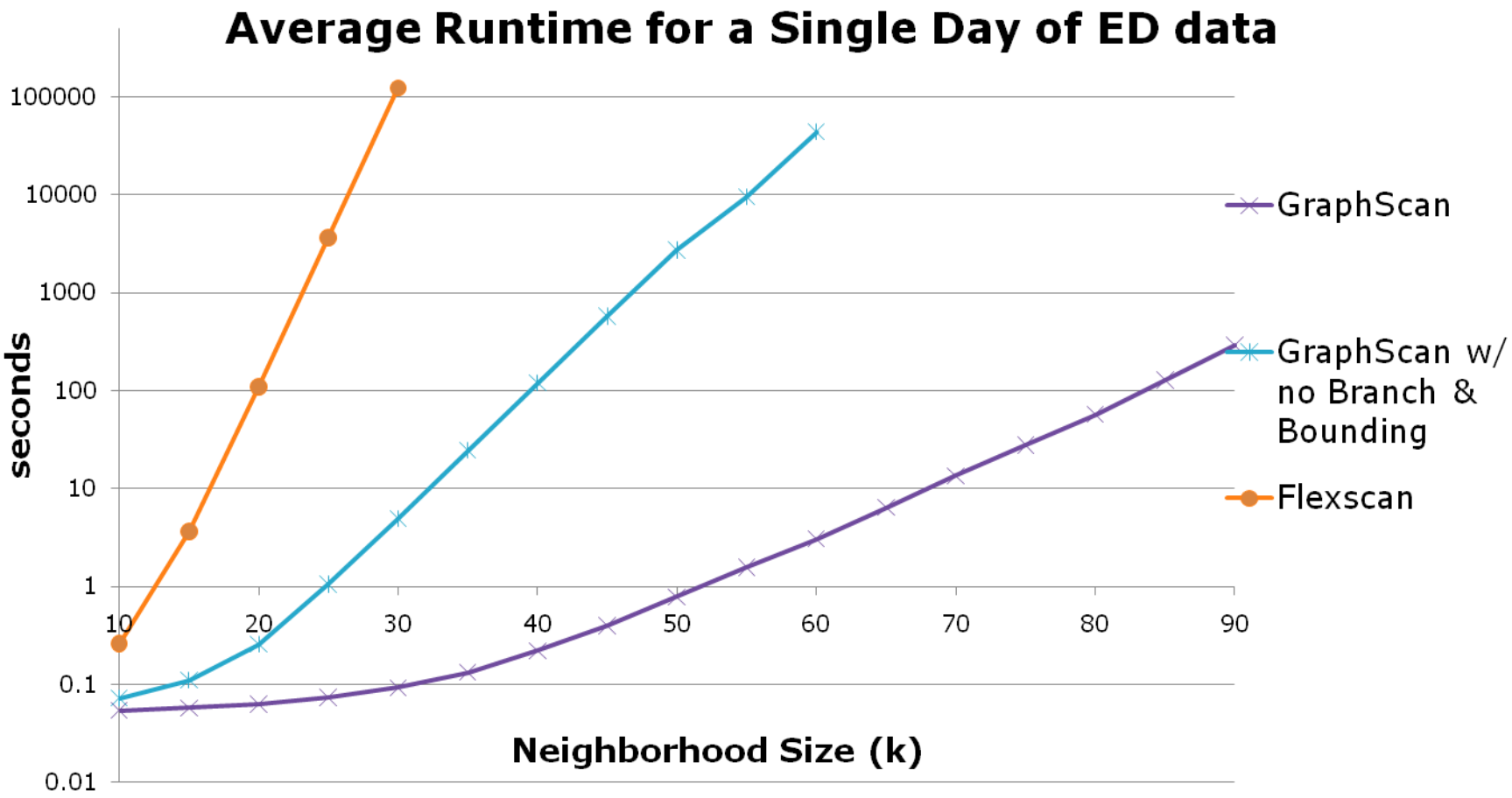
LTSS property with connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
and S \ {x} and S U {y} are both connected,
then subset S cannot be optimal."



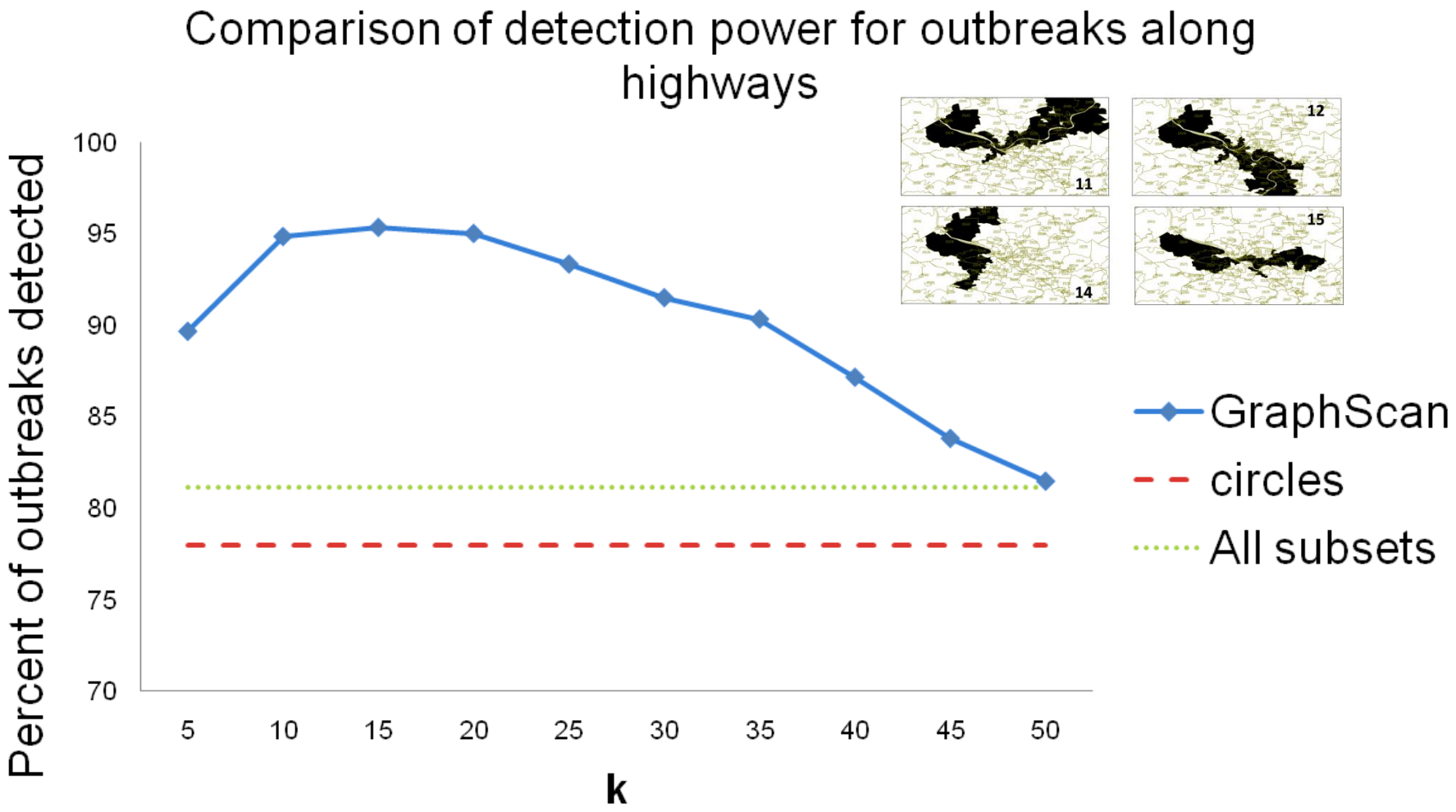
suboptimal

We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Evaluation: run times



Evaluation: detection power



Extensions of GraphScan

Q: What if we want to allow for events which spread dynamically over the graph structure?

A: Based on a new variant of the LTSS property, we can search for dynamic patterns while enforcing soft constraints on **temporal consistency.**

We have applied this method for more timely detection of contaminants spreading through a water distribution network.[1]

Q: What if the underlying graph structure is unknown?

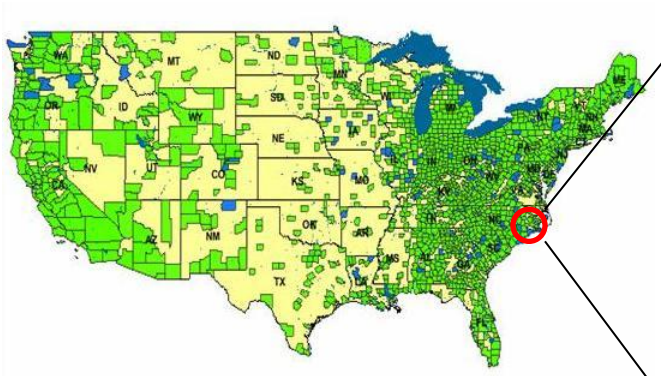
A: We can accurately **learn** the graph structure from unlabeled data, and use the learned structure for detection.

Often, the learned graph enables even faster detection of events than the true graph![2]

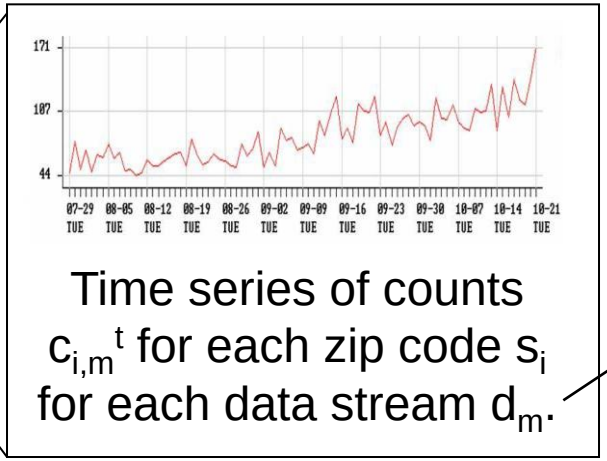
[1]Speakman and Neill, in preparation.

[2]Somanchi and Neill, submitted for publication.

Multidimensional event detection



Spatial time series data from spatial locations s_i (e.g. zip codes)



Time series of counts $c_{i,m}{}^t$ for each zip code s_i for each data stream d_m.

<u>Outbreak detection</u>

d_1 = respiratory ED

d_2 = constitutional ED

d_3 = OTC cough/cold

d_4 = OTC anti-fever

(etc.)

<u>Main goals</u>:

Detect any emerging events.

Pinpoint the affected subset of locations and time duration.

Characterize the event by identifying the affected streams.

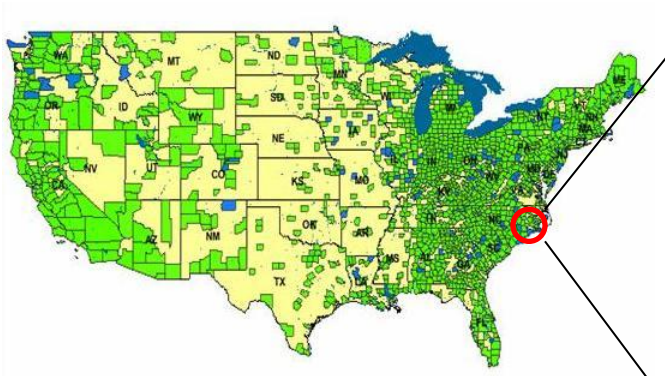
<u>Compare hypotheses</u>:

$H_1(D, S, W)$

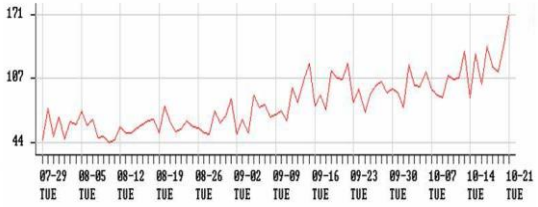
D = subset of streams
S = subset of locations
W = time duration

vs. H_0: no events occurring

Multidimensional event detection



Spatial time series data from spatial locations s_i (e.g. zip codes)



Time series of counts $c_{i,m}{}^t$ for each zip code s_i for each data stream d_m.

<u>Outbreak detection</u>

d_1 = respiratory ED

d_2 = constitutional ED

d_3 = OTC cough/cold

d_4 = OTC anti-fever

(etc.)

<u>Additional goal</u>: identify any differentially affected **subpopulations** P of the monitored population.

Gender (male, female, both)

Age groups (children, adults, elderly)

Ethnic or socio-economic groups

Risk behaviors: e.g. intravenous drug use, multiple sexual partners

More generally, assume that we have a set of additional discrete-valued attributes $A_1..A_J$ observed for each individual case.

We identify not only the affected streams, locations, and time window, but also a **subset** of values for each attribute.

Multidimensional LTSS

- Our **MD-Scan** approach (Neill and Kumar, 2013) extends MLTSS to the multidimensional case:
  - For each time window and spatial neighborhood (center + k-nearest neighbors), we do the following:

1. Start with randomly chosen subsets of **locations** S, **streams** D, and **values** V_j for each attribute A_j (j=1..J).

2. Choose an attribute (randomly or sequentially) and use LTSS to find the highest scoring subset of values, locations, or streams, conditioned on all other attributes.

3. Iterate step 2 until convergence to a local maximum of the score function F(D,S,W, $\{V_j\}$), and use multiple restarts to approach the global maximum.

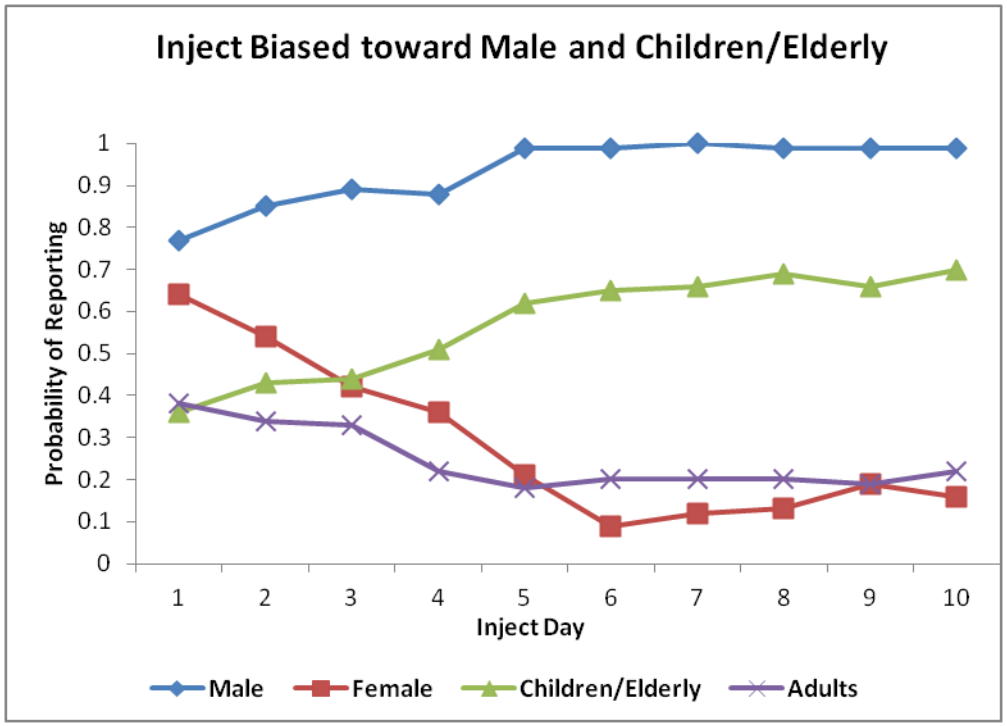
Evaluation

- We compared the detection performance of MD-Scan to MLTSS for detecting disease outbreaks injected into real-world Emergency Department data from Allegheny County, PA.
- For each case, the data included date, zip code, prodrome, gender, and age decile.
- We considered outbreaks with various types and amounts of age and gender bias.
  - Shown here: biased toward males, biased toward children and the elderly.

1) Identifying affected subpopulations

By the midpoint of the outbreak, MD-Scan is able to correctly identify the affected gender and age deciles with high probability, without reporting unaffected subpopulations.

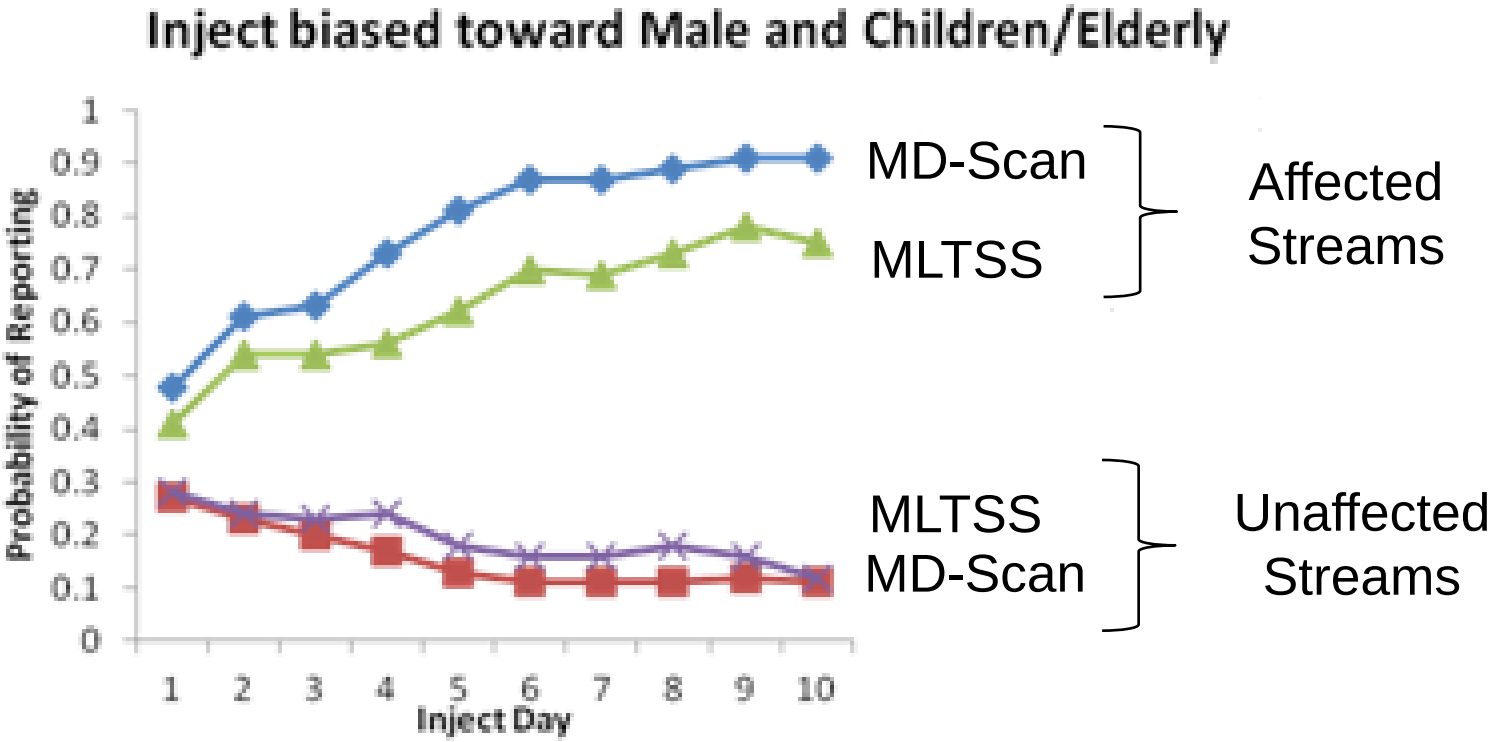
(MLTSS simply ignores the age and gender information, implicitly assuming that all ages and genders are affected.)



2) Characterizing affected streams

As compared to MLTSS, MD-Scan is better able to characterize the affected subset of the monitored streams.

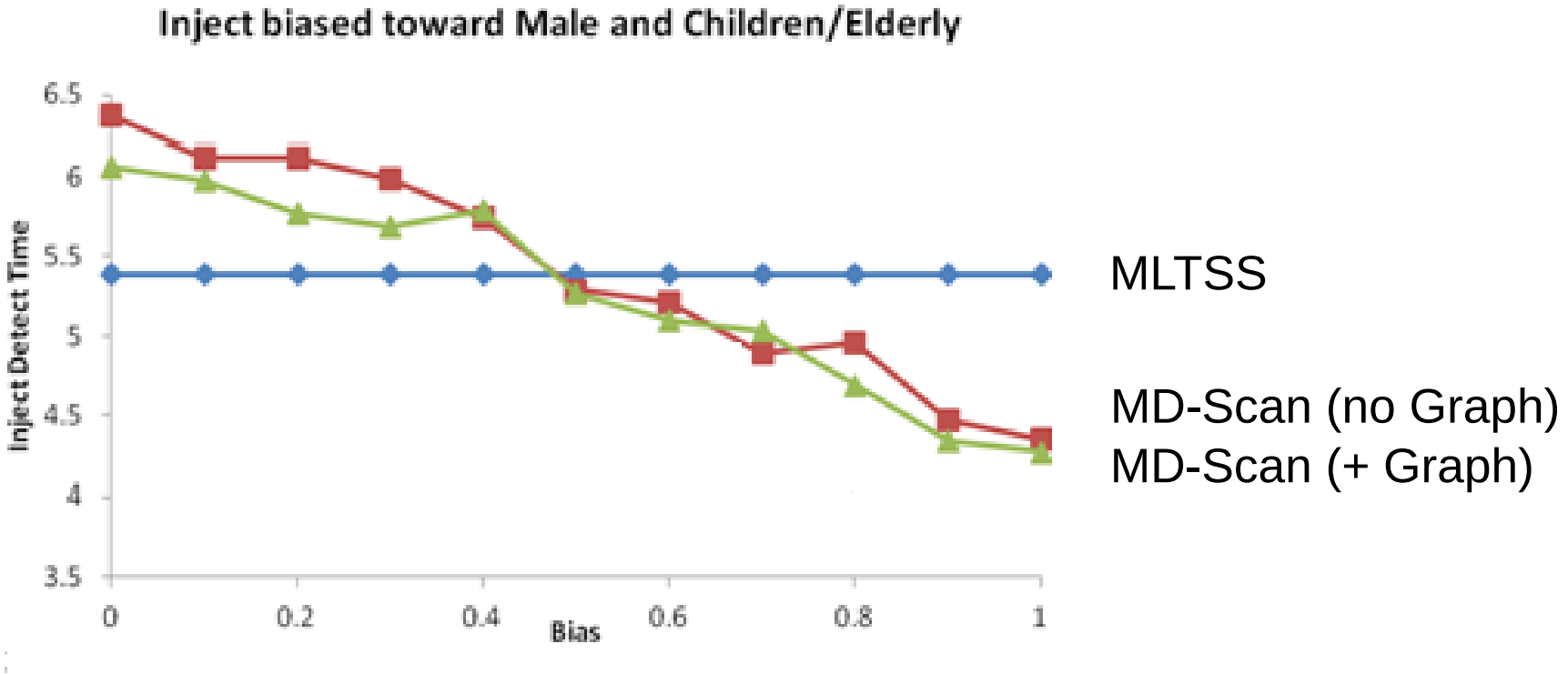
(Counts were injected into three of the eight monitored streams- respiratory, diarrhea, and fever.)



3) Time to detect (1 fp/month)

At a fixed false positive rate of 1 per month, MD-Scan achieved faster detection than MLTSS for outbreaks which were sufficiently biased by age and/or gender.

(Bias is linearly scaled, from 0 = same age/gender distribution as background data to 1 = only males and children/elderly affected.)



Current application domains

Biosurveillance: deployed systems in Ottawa, Grey-Bruce, Sri Lanka, India.

In progress: deployments in Canada for monitoring hospital-acquired illness, and patterns of harm related to drug abuse.

Crime prediction in Chicago:

Able to predict about 60% of "clustered" violent crimes with 15% false positive rate; also being applied to predicting citizen needs via 311 calls.

Many more applications:

Illicit container shipments
Clusters of water pipe breaks
Spreading water contamination
Network intrusion detection
Economic growth "outbreaks"
Conflict, violence, human rights

Detecting anomalous patterns of care in UPMC hospitals:

Our goal is to find atypical treatment conditions that improve patient outcomes ("best practices") or harm patients (systematic errors, improper hygiene, etc.)

References

- Daniel B. Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society (Series B: Statistical Methodology)* 74(2): 337-360, 2012.
- D. B. Neill, E. McFowland III, and H. Zheng. Fast subset scan for multivariate event detection. *Statistics in Medicine*, in press, 2013.
- D.B. Neill and T. Kumar. Fast multidimensional subset scan for outbreak detection and characterization. *Online Journal of Public Health Informatics* 5(1), 2013.
- S. Speakman, E. McFowland III, and D. B. Neill. Scalable detection of anomalous patterns with connectivity constraints. Submitted for publication.
- S. Speakman and D. B. Neill. Fast graph scan for scalable detection of arbitrary connected clusters. *Proc. ISDS Annual Conference*, 2009.
- D. B. Neill. Fast and flexible outbreak detection by linear-time subset scanning. *Advances in Disease Surveillance* 5:48, 2008.
- H. S. Burkom. Biosurveillance applying scan statistics with multiple disparate data sources. *J. Urban Health* 80: i57-i65, 2003.
- M. Kulldorff, F. Mostashari, L. Duczmal, K. Yih, K. Kleinman, and R. Platt. Multivariate spatial scan statistics for disease surveillance. *Statistics in Medicine* 26:1824-1833, 2007.

Interested?

More details on my web page:
http://www.cs.cmu.edu/~neill

Or e-mail me at:
neill@cs.cmu.edu