Approximate Statistical Discrepancy

Michael Matheny

with Jeff Phillips

School of Computing
University of Utah

December 17, 2018
Problem Setup

Let $X$ be a set of $m$ points in $\mathbb{R}^d$. 
Let $X$ be a set of $m$ points in $\mathbb{R}^d$.

Let $(X, \mathcal{A})$ be a range space, defining a family subsets $\mathcal{A}$ of $X$. 
Let $X$ be a set of $m$ points in $\mathbb{R}^d$.

Let $(X, \mathcal{A})$ be a range space, defining a family subsets $\mathcal{A}$ of $X$.

Each data point can belong to an anomalous set.
Let $X$ be a set of $m$ points in $\mathbb{R}^d$.

Let $(X, \mathcal{A})$ be a range space, defining a family subsets $\mathcal{A}$ of $X$.

Each data point can belong to an anomalous set.

Find a range $A \in \mathcal{A}$ where the anomalous data is significantly denser than the baseline data using some function $\Phi$. 
Statistical Functions Φ

2 scalar values for each $x \in X$: $b(x)$ and $r(x)$. For any $A \cap X$, define $B = \sum_{x \in X} b(x)$ and $R = \sum_{x \in X} r(x)$ and

$$b(A) = \frac{1}{B} \sum_{x \in X \cap A} b(x) \quad \text{and} \quad r(A) = \frac{1}{R} \sum_{x \in X \cap A} r(x)$$

Define statistic $\Phi(A)$ as the log likelihood ratio

$$\Phi(A) = \log \left( \frac{\Pr(\mathcal{H}_0|A, X)}{\Pr(\mathcal{H}_1|A, X)} \right)$$

- $\mathcal{H}_0$: no anomaly, rate of measured points same inside as outside.
- $\mathcal{H}_1$: anomaly, $A$ has a different rate of measured points inside than outside.
Anomaly Detection Pipeline

- Formulate a model of the data and choose a corresponding scan statistic $\phi$ to score the likelihood of an anomaly in a region.
Anomaly Detection Pipeline

- Formulate a model of the data and choose a corresponding scan statistic $\phi$ to score the likelihood of an anomaly in a region.
- Scan the data set to find a region $A^*$ which maximizes $\phi$. 
Anomaly Detection Pipeline

- Formulate a model of the data and choose a corresponding scan statistic $\phi$ to score the likelihood of an anomaly in a region.
- Scan the data set to find a region $A^*$ which maximizes $\phi$.
- Assess whether the score $\phi(A^*)$ indicates $A^*$ is significant.
▶ Formulate a model of the data and choose a corresponding scan statistic $\phi$ to score the likelihood of an anomaly in a region.
▶ Scan the data set to find a region $A^*$ which maximizes $\phi$.
▶ Assess whether the score $\phi(A^*)$ indicates $A^*$ is significant. Requires typically 1000 “scans” for permutation tests.
Anomaly Detection Pipeline

- Formulate a model of the data and choose a corresponding scan statistic $\phi$ to score the likelihood of an anomaly in a region.
- **Scan the data set to find a region $A^*$ which maximizes $\phi$.**
- Assess whether the score $\phi(A^*)$ indicates $A^*$ is significant. Requires typically 1000 “scans” for permutation tests.
Many existing papers on these algorithms:

- Classic discrepancy maximization [BDT16, DE93]
- Subroutine in algorithms ranging from computer graphics [DEM96] to association rules in data mining [FMMT96]
- Minimum disagreement problem in machine learning [LM96]
- Scan Statistics [Kul97, Kul06, HKG07, NM04, APV06, AMP⁺06, KHPD06, TT05](many many more)
Spatial Scan Statistics

Spatial Scan Statistics are heavily used to find spatial anomalies.

FiveThirtyEight

Politics  Sports  Science & Health  Economics  Culture

NOV. 16, 2018 AT 8:00 AM

How New York Hunts For Early Signs Of Disease Outbreaks

By Ian Evans  Filed under Public Health

On July 29, 2015, the New York City Department of Health and Mental Hygiene sent out an alert — 31 people in the South Bronx had contracted Legionnaires' disease, a lung infection from waterborne bacteria that kills about 1 out of every 10 people who get it. By the time officials found the source (a cooling tower) and contained the spread, 128 people had contracted Legionnaires' and 12 people had died. It was the largest outbreak of Legionnaires' disease in the city's history — an outbreak that was first detected by a computer program.
Approximate Problem

Find an approximate range $\hat{A} \in \mathcal{A}$ such that $\Phi(A^*) - \Phi(\hat{A}) < \varepsilon$. 
Approximate Problem

Find an approximate range $\hat{A} \in A$ such that $\Phi(A^*) - \Phi(\hat{A}) < \varepsilon$.

Existing approximation papers.

▶ [AMP$^+$06] which introduced generic sampling bounds and a bound on approximating scan statistics with linear functions.

▶ [MSZ$^+$16] which showed that a two-stage random sampling can provide some error guarantees.

▶ [Wal10] which showed approximation guarantees under the Bernoulli model.
Find an approximate range $\hat{A} \in \mathcal{A}$ such that $\Phi(A^*) - \Phi(\hat{A}) < \varepsilon$.

For $m = |X|$

<table>
<thead>
<tr>
<th>Range Space</th>
<th>Known Exact</th>
<th>Known Approx [MSZ\textsuperscript{+}16]</th>
<th>New Runtime Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Range Space</td>
<td>$O(m^{\nu+1})$</td>
<td>$O(m + \frac{1}{\varepsilon^4} \log^{\nu} \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^{\nu+2}} \log^{\nu} \frac{1}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Halfspaces</td>
<td>$O(m^d)$</td>
<td>$O(m + \frac{1}{\varepsilon^d} \log^{\nu} \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^{d+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Disks</td>
<td>$O(m^3)$</td>
<td>$O(m + \frac{1}{\varepsilon^4} \log^{3} \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^{3+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Rectangles (Disc)</td>
<td>$O(m^2)$</td>
<td>$O(m + \frac{1}{\varepsilon^4} \log \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Rectangles (sdf)</td>
<td>$O(m^4)$</td>
<td>$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Rectangles (Gen)</td>
<td>$O(m^4)$</td>
<td>$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$</td>
<td>$O\left(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon}\right)$</td>
</tr>
</tbody>
</table>

Algorithm times for ($\varepsilon$-approximately) maximizing different range spaces. Here dimension $d$, VC-dimension $\nu$, and probability of failure are all constants. For Rectangles (Disc) we show it takes $\Omega(m + 1/\varepsilon^2)$ time, assuming hardness of APSP.
In some case linear $\phi$ functions are easier:

$$\phi(r, b) = C_1 r + C_2 b$$
Approximating $\phi$

In some cases, linear $\phi$ functions are easier:

$$\phi(r, b) = C_1 r + C_2 b$$

- Agarwal [AMP$^+$06] showed can be $\varepsilon$-approximated with lower envelope of $O(1/\varepsilon \log \frac{1}{\varepsilon})$ linear functions.

- New result: only need $O(1/\sqrt{\varepsilon})$ linear functions.

In practice, only need 3 to 4 linear functions.
Approximating $\phi$

In some cases, linear $\phi$ functions are easier:

$$\phi(r, b) = C_1 r + C_2 b$$

- Agarwal [AMP+06] showed can be $\varepsilon$-approximated with lower envelope of $O(1/\varepsilon \log \frac{1}{\varepsilon})$ linear functions.
- New result: only need $O(1/\sqrt{\varepsilon})$ linear functions.
In some case linear $\phi$ functions are easier:

$$\phi(r, b) = C_1 r + C_2 b$$

- Agarwal [AMP$^+$$06]$ showed can be $\varepsilon$-approximated with lower envelope of $O(1/\varepsilon \log \frac{1}{\varepsilon})$ linear functions.
- New result: only need $O(1/\sqrt{\varepsilon})$ linear functions.
- In practice only need 3 to 4 linear functions.
Approximate $\phi$ with $O(1/\sqrt{\varepsilon})$ linear functions.

- If $\phi$ is a convex function then $\max$ lies on convex hull of:

$$V_{A,m,b} = \{(r(A), b(A)) \mid A \in \mathcal{A}\}$$

- Approximate convex hull of $V_{A,r,b}$ by picking linear functions in an iterative way.
Approximating $\phi$ (cont)

Approximate $\phi$ with $O(1/\sqrt{\varepsilon})$ linear functions.

- If $\phi$ is a convex function then max lies on convex hull of:

$$V_{A,m,b} = \{(r(A), b(A)) \mid A \in \mathcal{A}\}$$

- Approximate convex hull of $V_{A,r,b}$ by picking linear functions in an iterative way.

- Size bound by Dudley’s approximation ($\varepsilon$-kernel)
Approximate Rectangle Scanning

- Grid $G$ over $X$ so that each row and column has $\approx \varepsilon |X|$ points.

![Diagram of grid over $X$ with points approximated]
Grid $G$ over $X$ so that each row and column has $\approx \epsilon |X|$ points.

Each rectangle $R \in (X, \mathcal{R}_2)$ is approximated by a subgrid $R_G \in G$. 
Approximate Rectangle Scanning

- Grid $G$ over $X$ so that each row and column has $\approx \varepsilon |X|$ points.
- Each rectangle $R \in (X, \mathcal{R}_2)$ is approximated by a subgrid $R_G \in G$.
- Can enumerate all subgrids in $O(\frac{1}{\varepsilon^4})$ time and compute $\Phi$ on each. ($O(m + \frac{1}{\varepsilon^4})$)
Approximate Rectangle Scanning for Linear Function

- Grid $G$ over $X$ so that each row and column has $\approx \varepsilon |X|$ points.
- Fix $1/\varepsilon$ upper end-points and sweep $1/\varepsilon$ lower end-points.

![Diagram of grid and end-points]
Approximate Rectangle Scanning for Linear Function

- Grid $G$ over $X$ so that each row and column has $\approx \varepsilon |X|$ points.
- Fix $1/\varepsilon$ upper end-points and sweep $1/\varepsilon$ lower end-points.
- Run Kadane’s algorithm for each upper and lower endpoint to compute maximum horizontal subgrid in time $O(\frac{1}{\varepsilon})$. 

![Diagram of grid and scanning process]
Approximate Rectangle Scanning for Linear Function

- Grid $G$ over $X$ so that each row and column has $\approx \varepsilon |X|$ points.
- Fix $1/\varepsilon$ upper end-points and sweep $1/\varepsilon$ lower end-points.
- Run Kadane's algorithm for each upper and lower endpoint to compute maximum horizontal subgrid in time $O\left(\frac{1}{\varepsilon}\right)$.
- Max subgrid in $O\left(\frac{1}{\varepsilon^3}\right)$ time (max likelihood function in $O\left(\frac{1}{\varepsilon^{3.5}}\right)$ time.)
Can be made faster:

- Previously scanning was exact and approximation was from restricting to a grid.
Faster Approximate Rectangle Scanning

Can be made faster:

▶ Previously scanning was exact and approximation was from restricting to a grid.

▶ Most subgrids are very similar.
Can be made faster:

- Previously scanning was exact and approximation was from restricting to a grid.
- Most subgrids are very similar.
- Make scanning approximate so that only small updates occur to similar subgrids.
Consider computing the max subgrid spanning a slab $M$. 

- Divide upper subgrid into subgrids $T_T$ and $T_B$ and lower subgrids into $B_T$ and $B_B$.
- Decompose into 4 separate problems.

Idea inspired by [BCNPL14, Tak02, DEM96]
Consider computing the max subgrid spanning a slab $M$.
Divide upper subgrid into subgrids $T_T$ and $T_B$ and lower subgrids into $B_T$ and $B_B$.
Decompose into 4 separate problems.
Consider computing the max subgrid spanning a slab $M$.

Divide upper subgrid into subgrids $T_T$ and $T_B$ and lower subgrids into $B_T$ and $B_B$.

Decompose into 4 separate problems.
Faster Approximate Rectangle Scanning

- Consider computing the max subgrid spanning a slab $M$.
- Divide upper subgrid into subgrids $T_T$ and $T_B$ and lower subgrids into $B_T$ and $B_B$.
- Decompose into 4 separate problems.

Idea inspired by [BCNPL14, Tak02, DEM96]
Consider computing the max subgrid spanning a slab $M$.

Divide upper subgrid into subgrids $T_T$ and $T_B$ and lower subgrids into $B_T$ and $B_B$.

Decompose into 4 separate problems.

Idea inspired by [BCNPL14, Tak02, DEM96]
Have to merge $T_B$ and/or merge $B_T$ into $M$.
1. Merge neither into $M$.
2. Merge $T_B$ into $M$.
3. Merge $B_T$ into $M$.
4. Merge $T_B$ and $B_T$ into $M$. 
Faster Approximate Rectangle Scanning

Merging can be done in time proportional to non-zeros columns (see paper for details).

- If $T_b$ or $B_b$ has $k$ rows then can construct sparse grid with $O(k \log \frac{1}{\varepsilon})$ non zero columns that misplaces $\varepsilon/\log \frac{1}{\varepsilon} |X|$ points with respect to vertical intervals.
Merging can be done in time proportional to non-zeros columns (see paper for details).

- If $T_b$ or $B_b$ has $k$ rows then can construct sparse grid with $O(k \log \frac{1}{\varepsilon})$ non zero columns that misplaces $\varepsilon/\log \frac{1}{\varepsilon} |X|$ points with respect to vertical intervals.
Have to construct a tree of sparse *subgrids* first.

Error adds over $\log \frac{1}{\varepsilon}$ levels in the recurrence leading to $\varepsilon |X|$ misplaced points.

Total run time is $O(m + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$

- $O(m + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$ time to build grid.
- Takes $O(\frac{1}{\varepsilon^2})$ time to construct tree of sparse subgrids.
- Takes $O(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$ to compute max slab spanning subgrid approximately.

$log \frac{1}{\varepsilon}$ can be made $log \log \frac{1}{\varepsilon}$ with some work.

New lower bound conditional on APSP of $\Omega(|X| + \frac{1}{\varepsilon^2})$. 
Faster in Practice

Significant improvement in convergence

- SatScan is exact algorithm run on sample.
- gridScan is simple scanning over a grid.
- gridScan_linear is Kadane based algorithm.
- Do not have an implementation of fastest algorithm.
Other range spaces
2 Level Scanning

Why are exact algorithms slow on a Sample?
2 Level Scanning

Problem: Far too many combinatorial regions.
2 Level Scanning

Idea: Use smaller sample of size $O_d(\frac{1}{\varepsilon})$ to induce regions.
2 Level Scanning

Idea: Use smaller sample of size $O_d(\frac{1}{\varepsilon})$ to induce regions.
2 Level Scanning

Compute $\phi$ using dense sample.
Repeat procedure $\log \frac{1}{\delta}$ times and take median to amplify probability of success.
2 Level Scanning

Procedure works on many different range spaces.
Dobkin and Eppstein [DE93] maximize \((X, \mathcal{H})\) of \(n\) points in \(\mathbb{R}^d\) in \(O(n^d)\) time.

**Primal**

**Dual**
Dobkin and Eppstein [DE93] maximize $(X, \mathcal{H})$ of $n$ points in $\mathbb{R}^d$ in $O(n^d)$ time.
Dobkin and Eppstein [DE93] maximize \((X, \mathcal{H})\) of \(n\) points in \(\mathbb{R}^d\) in \(O(n^d)\) time. But need to count points in \(S\).

Can annotate dual arrangement in \(O(n)\) time in \(\mathbb{R}^2\) for each \(x \in S\).
Faster with better coreset bounds?

$S$ is usually of size $O\left(\frac{1}{\varepsilon^2}\right)$ for constant $d$

- Halfspaces $A = \mathcal{H}$ in $\mathbb{R}^2$ then
  \[ |S| = s = O((1/\varepsilon)^{4/3}) \]
- Balls $A = \mathcal{B}$ in $\mathbb{R}^2$ then
  \[ |S| = s = O((1/\varepsilon)^{4/3}) \]

Provides further speedup.
Faster with better coreset bounds?

$S$ is usually of size $O\left(\frac{1}{\varepsilon^2}\right)$ for constant $d$

- Halfspaces $A = \mathcal{H}$ in $\mathbb{R}^2$ then
  $|S| = s = O((1/\varepsilon)^{4/3})$

- Balls $A = \mathcal{B}$ in $\mathbb{R}^2$ then
  $|S| = s = O((1/\varepsilon)^{4/3})$

Method exists for computing Halfspaces samples [MP18]:

- $|S| = O((1/\varepsilon)^{2d/(d+1)} \log^{d/(d+1)} (1/\varepsilon))$

- Computable in $O(n + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$ time.

Provides further speedup.
### Summary

For $m = |X|$  

<table>
<thead>
<tr>
<th></th>
<th>Known Exact</th>
<th>Known Approx [MSZ\textsuperscript{+}16]</th>
<th>New Runtime Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Range Space</td>
<td>$O(m^{\nu+1})$</td>
<td></td>
<td>$O \left(m + \frac{1}{\epsilon\nu+2} \log^{\nu} \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>Halfspaces</td>
<td>$O(m^d)$ [DEM96]</td>
<td></td>
<td>$O \left(m + \frac{1}{\epsilon d+1/3} \log^{2/3} \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>Disks</td>
<td>$O(m^3)$ [DEM96]</td>
<td></td>
<td>$O \left(m + \frac{1}{\epsilon^3+1/3} \log^{2/3} \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>Rectangles (Disc)</td>
<td>$O(m^2)$ [BCNPL14]</td>
<td>$O(m + \frac{1}{\epsilon^4} \log \frac{1}{\epsilon})$</td>
<td>$O \left(m + \frac{1}{\epsilon^2} \log \log \frac{1}{\epsilon} \right)$</td>
</tr>
<tr>
<td>Rectangles (SDF)</td>
<td>$O(m^4)$</td>
<td>$O(m + \frac{1}{\epsilon^4} \log^4 \frac{1}{\epsilon})$</td>
<td>$O \left(m + \frac{1}{\epsilon^{2.5}} \right)$</td>
</tr>
<tr>
<td>Rectangles (Gen)</td>
<td>$O(m^4)$</td>
<td>$O(m + \frac{1}{\epsilon^4} \log^4 \frac{1}{\epsilon})$</td>
<td>$O \left(m + \frac{1}{\epsilon^{4}} \right)$</td>
</tr>
</tbody>
</table>

Algorithm times for ($\epsilon$-approximately) maximizing different range spaces. Here dimension $d$, VC-dimension $\nu$, and probability of failure are all constants. For Rectangles (Disc) we show it takes $\Omega(m + 1/\epsilon^2)$ time, assuming hardness of APSP.

Any Questions?


For Further Reading III


Evaluating Discrepancy

\[ b(A) = \frac{1}{B} \sum_{x \in X \cap A} b(x) \quad \text{and} \quad r(A) = \frac{1}{R} \sum_{x \in X \cap A} r(x) \]

Let \( b(x) = -1 \) and \( m(x) = \{0, +2\} \)

\[ \phi(b, r) = |B(m(A) + b(A))| \]
Evaluating Discrepancy

\[ \sum_{x \in X \cap A} b(x) \quad \text{and} \quad \sum_{x \in X \cap A} r(x) \]

Let \( b(x) = -1 \) and \( m(x) = \{0, +2\} \)

\[ \phi(b, r) = |B(m(A) + b(A))| \]

Equivalent to \textit{discrepancy evaluation} for a range space \((X, A)\) and a coloring \( \chi : X \rightarrow \{-1, +1\} \): Find

\[ \text{disc}_\chi(X, A) = \arg \max_{A \in \mathcal{A}} \left| \sum_{x \in X \cap A} \chi(x) \right| \]

If \( \chi(x) = +1 \) then \( r(x) = 2 \) otherwise if \( \chi(x) = -1 \) then \( r(x) = 0 \).