Approximate Statistical Discrepancy

Michael Matheny

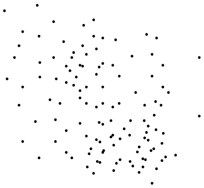
with Jeff Phillips

School of Computing
University of Utah

December 17, 2018

Problem Setup

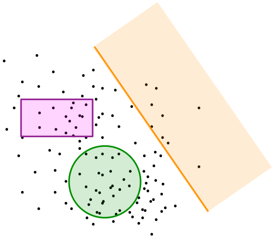
Let X be a set of m points in $\mathbb{R}^d$.



Problem Setup

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.

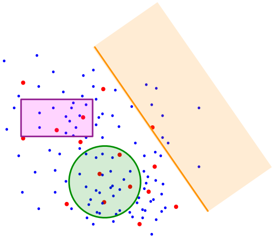


Problem Setup

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



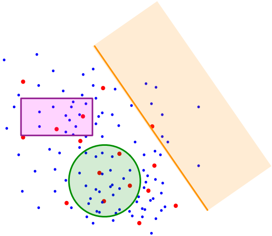
Problem Setup

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 Φ



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 ϕ 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 ϕ to score the likelihood of an anomaly in a region.
- Scan the data set to find a region A^* which maximizes ϕ.

Anomaly Detection Pipeline

- Formulate a model of the data and choose a corresponding scan statistic ϕ to score the likelihood of an anomaly in a region.
- Scan the data set to find a region A^* which maximizes ϕ.
- Assess whether the score $\phi(A^*)$ indicates A^* is significant.

Anomaly Detection Pipeline

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

Existing Work

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, 2016 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 $\hat{A} \in \mathcal{A}$ 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.

Approximate Problem

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

For $m = |X|$

	Known Exact	Known Approx [MSZ+16]	**New Runtime Bounds**
General Range Space	$O(m^{\nu+1})$		$O\left(m + \frac{1}{\varepsilon^{\nu+2}} \log^{\nu} \frac{1}{\varepsilon}\right)$
Halfspaces	$O(m^d)$ [DEM96]	–	$O\left(m + \frac{1}{\varepsilon^{d+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Disks	$O(m^3)$ [DEM96]	$O(m + \frac{1}{\varepsilon^4} \log^3 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{3+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Rectangles (Disc)	$O(m^2)$ [BCNPL14]	$O(m + \frac{1}{\varepsilon^4} \log \frac{1}{\varepsilon})$ [APV06]	$O(m + \frac{1}{\varepsilon^2} \log\log \frac{1}{\varepsilon})$
Rectangles (SDF)	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{2.5}}\right)$
Rectangles (Gen)	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon}))$	$O\left(m + \frac{1}{\varepsilon^4}\right)$

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

Approximating ϕ

In some case linear ϕ functions are easier:

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

Approximating ϕ

In some case linear ϕ functions are easier:

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

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

Approximating ϕ

In some case linear ϕ functions are easier:

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

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

Approximating ϕ

In some case linear ϕ functions are easier:

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

- Agarwal [AMP+06] showed can be ε-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.

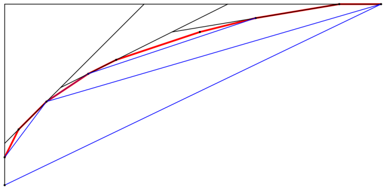
Approximating ϕ (cont)

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

- If ϕ is a convex function then max lies on convex hull of:

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

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



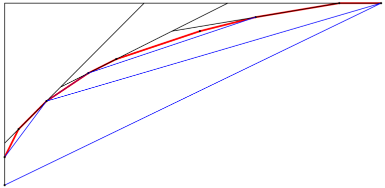
Approximating ϕ (cont)

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

- If ϕ is a convex function then max lies on convex hull of:

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

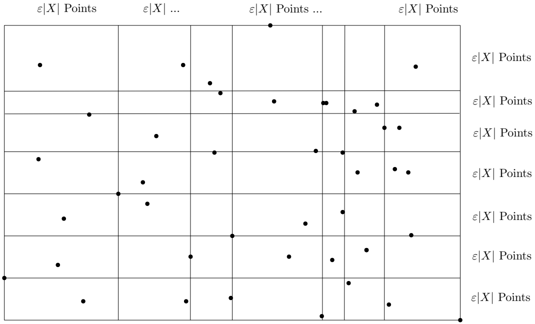
- Approximate convex hull of $V_{\mathcal{A},r,b}$ by picking linear functions in an iterative way.



- Size bound by Dudley's approximation (ε-kernel)

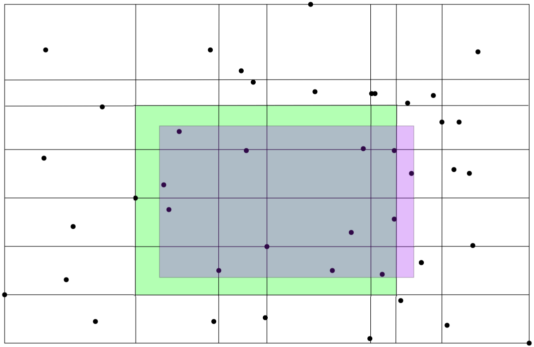
Approximate Rectangle Scanning

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



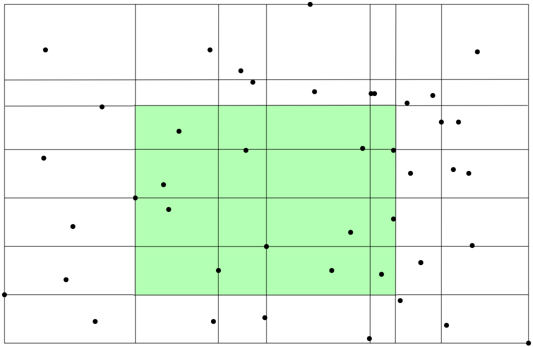
Approximate Rectangle Scanning

- 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 $R_G \in G$.



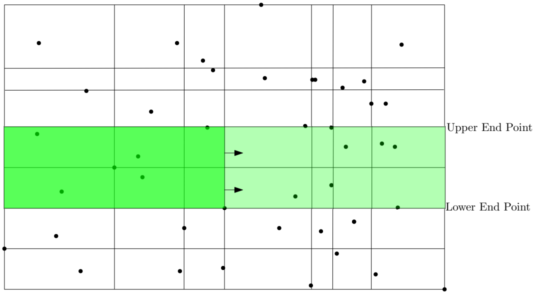
Approximate Rectangle Scanning

- 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 Φ on each.($O(m + \frac{1}{\varepsilon^4})$)



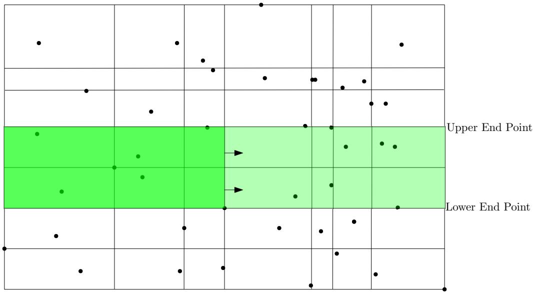
Approximate Rectangle Scanning for Linear Function

- 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.



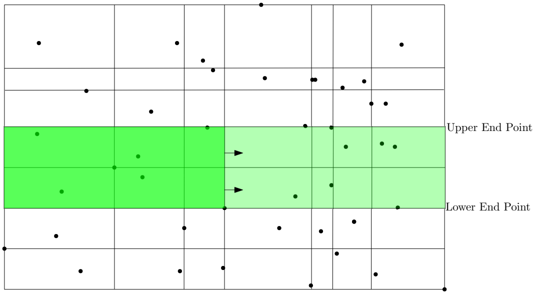
Approximate Rectangle Scanning for Linear Function

- 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 $O(\frac{1}{\varepsilon})$.



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 $O(\frac{1}{\varepsilon})$.
- Max subgrid in $O(\frac{1}{\varepsilon^3})$ time (max likelihood function in $O(1/\varepsilon^{3.5})$ time.)



Faster Approximate Rectangle Scanning

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.

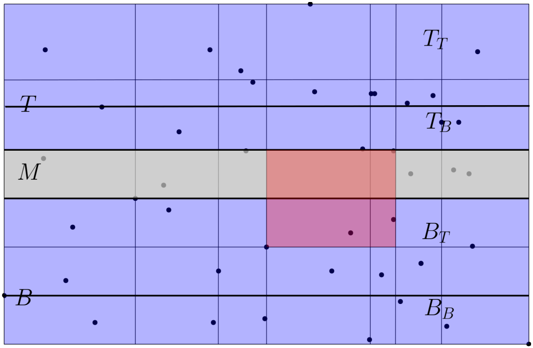
Faster Approximate Rectangle Scanning

Can be made faster:

- Previously scanning was exact and approximation was from restricting to a grid.
- Most subgrids are very similar.
- Make scanning approximate so that only small updates occur to similar subgrids.

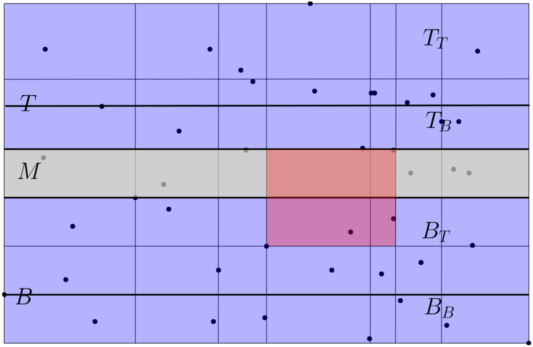
Faster Approximate Rectangle Scanning

- Consider computing the max subgrid spanning a slab M.



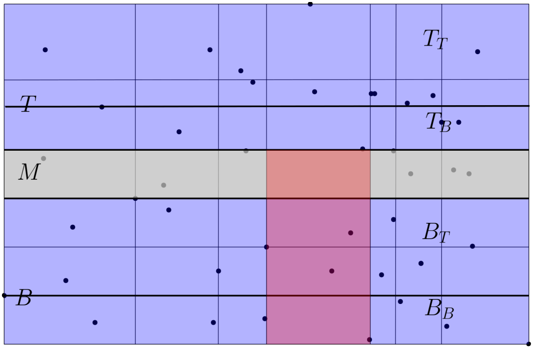
Faster Approximate Rectangle Scanning

- 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 seperate problems.



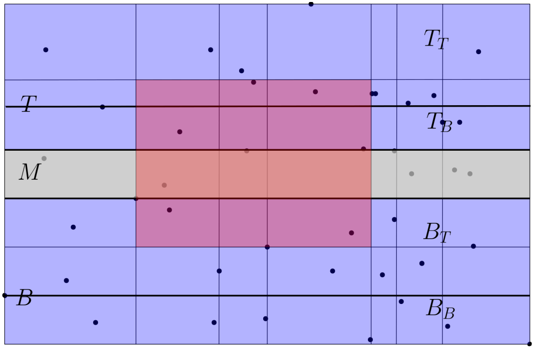
Faster Approximate Rectangle Scanning

- 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 seperate problems.



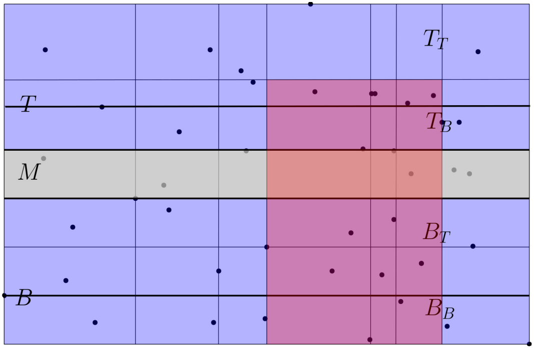
Faster Approximate Rectangle Scanning

- 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 seperate problems.



Faster Approximate Rectangle Scanning

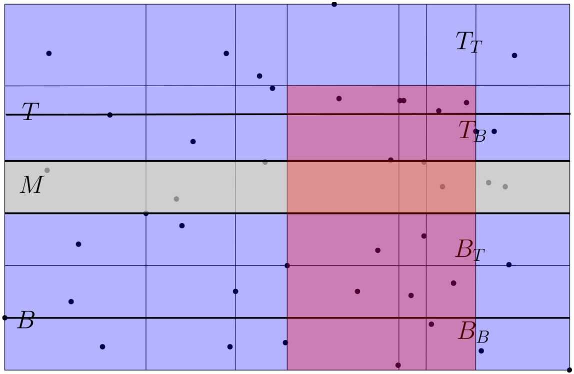
- Consider computing the max subgrid spanning a slab M.
- Divide upper subgrid into subgrids T_T and T_B and lower subgrids into B_T and B_B.
- Decompose into 4 seperate problems.
- Idea inspired by [BCNPL14, Tak02, DEM96]



Faster Approximate Rectangle Scanning

Have to merge T_B and/or merge B_T into M.

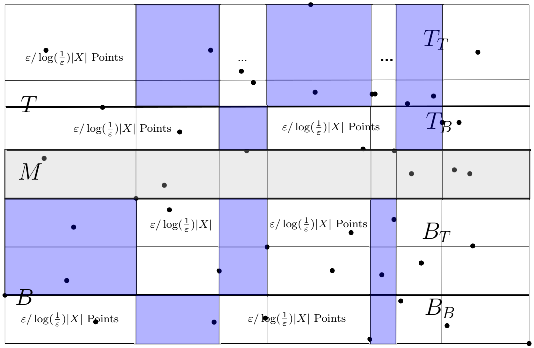
1. Merge neither into M.
2. Merge T_B into M.
3. Merge B_T into M.
4. Merge T_B and B_T into M.



Faster Approximate Rectangle Scanning

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

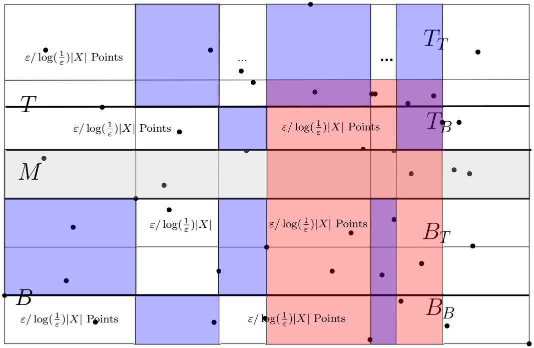
- If T_b or B_b has k rows then can construct sparse grid with $O(k \log \frac{1}{\varepsilon})$ non zero columns that misplaces $\varepsilon / \log \frac{1}{\varepsilon} |X|$ points with respect to vertical intervals.



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.



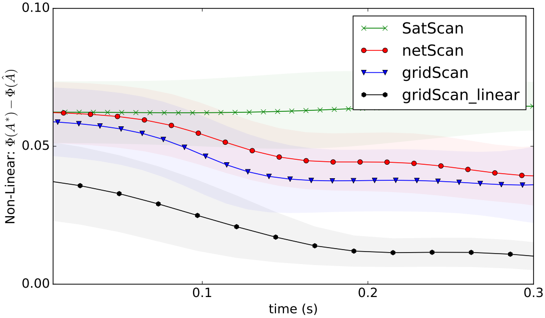
Faster Approximate Rectangle Scanning

- 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

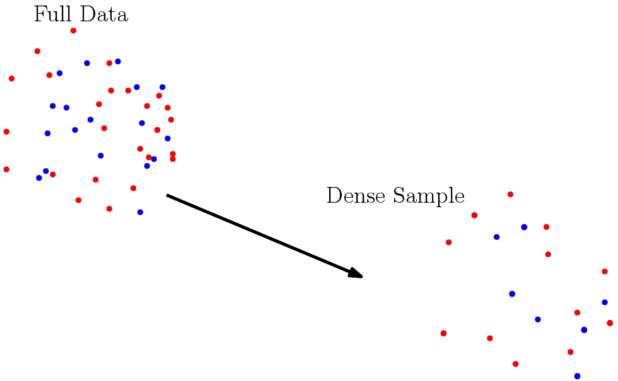
- SatScan is exact algorithm run on sample.
- gridScan is simple scanning over a grid.
- gridScan_linear is Kadane based algorithm.
- Do not have an implementation of fastest algorithm.



Other range spaces

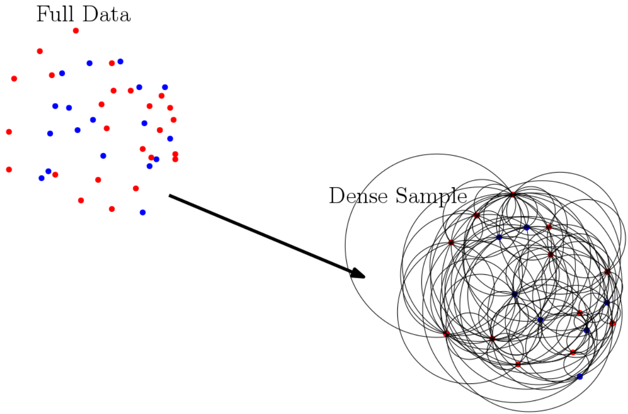
2 Level Scanning

Why are exact algorithms slow on a Sample?



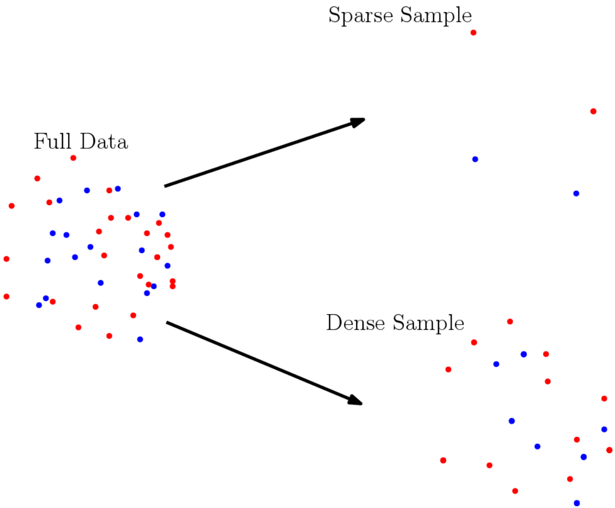
2 Level Scanning

Problem: Far too many combinatorial regions.



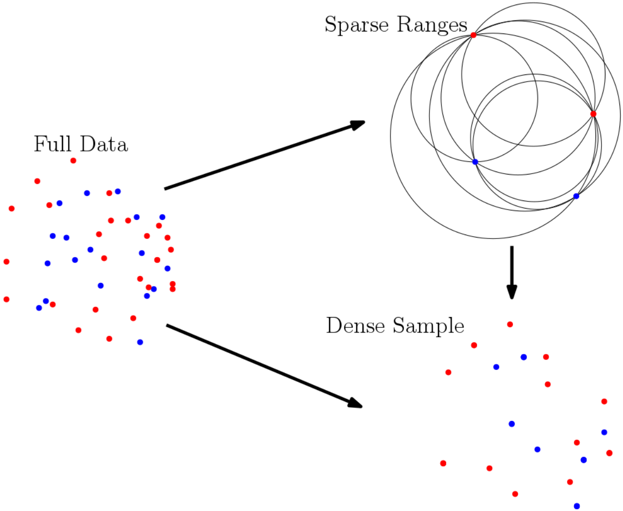
2 Level Scanning

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



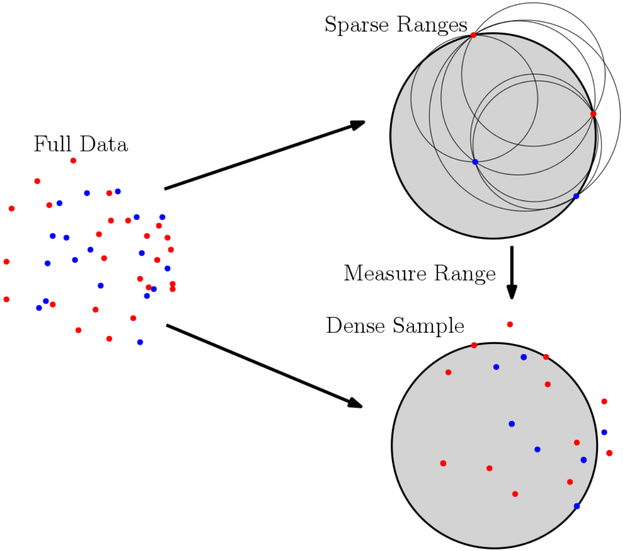
2 Level Scanning

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



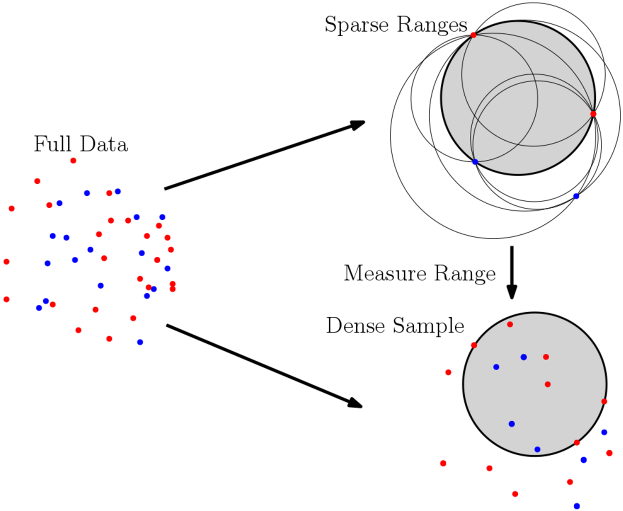
2 Level Scanning

Compute ϕ using dense sample.



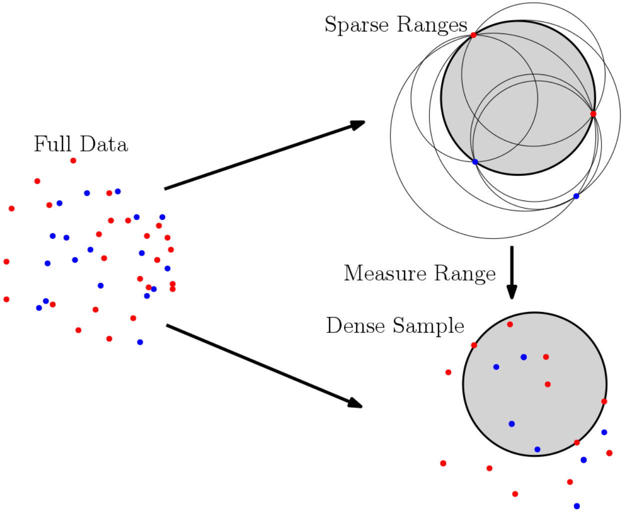
2 Level Scanning

Repeat procedure $\log \frac{1}{\delta}$ times and take median to amplify probability of success.



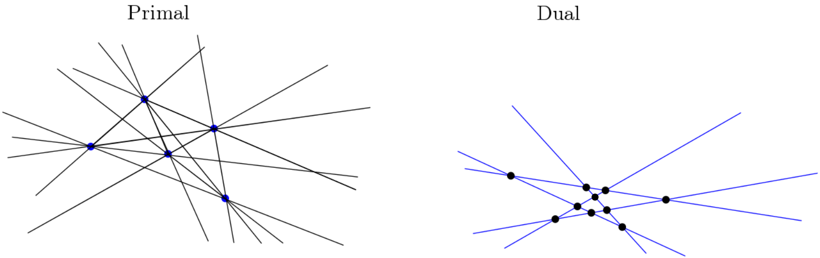
2 Level Scanning

Procedure works on many different range spaces.



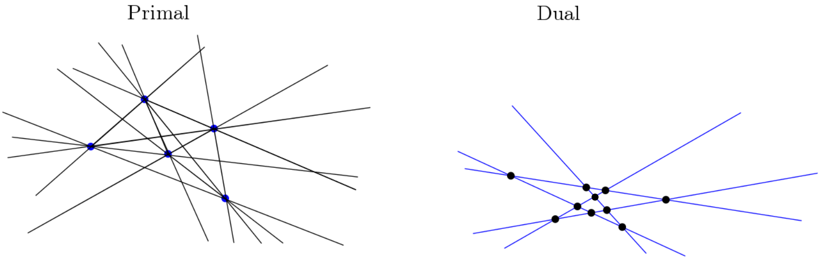
Fast Scanning of Halfspaces

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



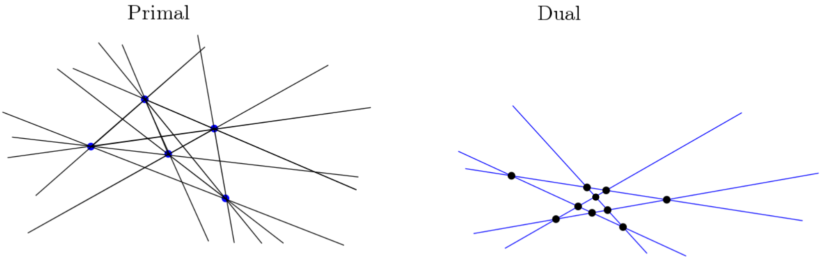
Fast Scanning of Halfspaces

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



Fast Scanning of Halfspaces

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 $O(\frac{1}{\varepsilon^2})$ for constant d

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

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

	Known Exact	Known Approx [MSZ+16]	**New Runtime Bounds**
General Range Space	$O(m^{\nu+1})$		$O\left(m + \frac{1}{\varepsilon^{\nu+2}} \log^{\nu} \frac{1}{\varepsilon}\right)$
Halfspaces	$O(m^d)$ [DEM96]	–	$O\left(m + \frac{1}{\varepsilon^{d+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Disks	$O(m^3)$ [DEM96]	$O(m + \frac{1}{\varepsilon^4} \log^3 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{3+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Rectangles (Disc)	$O(m^2)$ [BCNPL14]	$O(m + \frac{1}{\varepsilon^4} \log \frac{1}{\varepsilon})$ [APV06]	$O(m + \frac{1}{\varepsilon^2} \log\log \frac{1}{\varepsilon})$
Rectangles (SDF)	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{2.5}}\right)$
Rectangles (Gen)	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon}))$	$O\left(m + \frac{1}{\varepsilon^4}\right)$

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

Any Questions?

For Further Reading I

- Deepak Agarwal, Andrew McGregor, Jeff M. Phillips, Suresh Venkatasubramanian, and Zhengyuan Zhu, *Spatial scan statistics: Approximations and performance study*, KDD, 2006.

- Deepak Agarwal, Jeff M. Phillips, and Suresh Venkatasubramanian, *The hunting of the bump: On maximizing statistical discrepancy*, SODA (2006).

- Jérémy Barbay, Timothy M. Chan, Gonzalo Navarro, and Pablo Pérez-Lantero, *Maximum-weight planar boxes in time (and better)*, Information Processing Letters **114** (2014), no. 8, 437 – 445.

- Arturs Backurs, Nishanth Dikkala, and Christos Tzamos, *Tight hardness results for maximum weight rectangles*, ICALP, 2016.

- David Dobkin and David Eppstein, *Computing the discrepancy*, Proceedings 9th Annual Symposium on Computational Geometry, 1993.

- David P. Dobkin, David Eppstein, and Don P. Mitchell, *Computing the discrepancy with applications to supersampling patterns*, ACM Trans. Graph. **15** (1996), no. 4, 354–376.

For Further Reading II

- Takeshi Fukuda, Yasukiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama, *Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization*, SIGMOD Rec. **25** (1996), no. 2, 13–23.
- Lan Huang, Martin Kulldorff, and David Gregorio, *A spatial scan statistic for survival data*, BioMetrics **63** (2007), 109–118.
- Martin Kulldorff, Lan Huang, Linda Pickle, and Luiz Duczmal, *An elliptic spatial scan statistic.*, Statistics in medicine **25 22** (2006), 3929–43.
- Martin Kulldorff, *A spatial scan statistic*, Communications in Statistics: Theory and Methods **26** (1997), 1481–1496.
- ______, *Satscan user guide*, http://www.satscan.org/, 7.0 ed., 2006.
- Ming C Lin and Dinesh Manocha, *Applied computational geometry. towards geometric engineering: Selected papers*, vol. 114, Springer Science & Business Media, 1996.

For Further Reading III

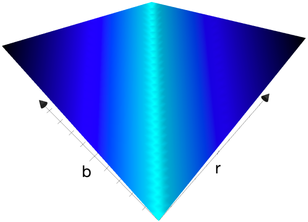
- Michael Matheny and Jeff M. Phillips, *Practical low-dimensional halfspace range space sampling*, European Symposium on Algorithms (arXiv:1804.11307), 2018.
- Michael Matheny, Raghvendra Singh, Liang Zhang, Kaiqiang Wang, and Jeff M. Phillips, *Scalable spatial scan statistics through sampling*, SIGSPATIAL, 2016.
- Daniel B. Neill and Andrew W. Moore, *Rapid detection of significant spatial clusters*, KDD, 2004.
- Tadao Takaoka, *Efficient algorithms for the maximum subarray problem by distance matrix multiplication*, CATS (2002).
- Toshiro Tango and Kunihiko Takahashi, *A flexibly shaped spatial scan statistic for detecting clusters*, International Journal of Health Geographics **4** (2005), no. 1, 11.
- Guenther Walther, *Optimal and fast detection of spatial clusters with scan statistics*, Ann. Statist. **38** (2010), no. 2, 1010–1033.

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

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

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

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

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

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

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