Wednesday April 26, 2017 12:00pm

Active Search with Complex Actions and Rewards

Yifei Ma

Thesis Defense with Committee:

Jeff Schneider, Roman Garnett, Aarti Singh, Alex Smola, Ryan P. Adams

Slides: http://yma.io/thesis_slides.pdf

Auton Lab

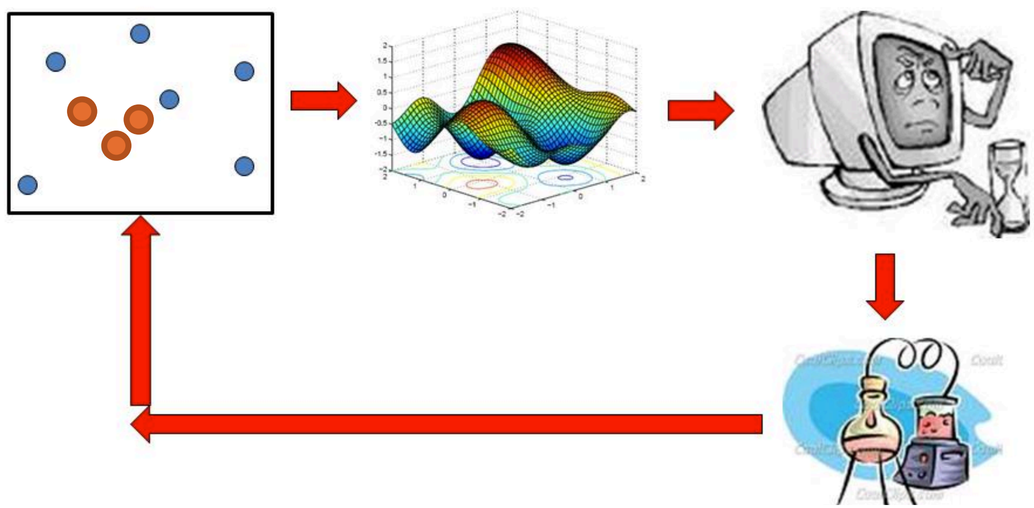
ML MACHINE LEARNING DEPARTMENT

Carnegie Mellon University

Active Search

Find all positives in an unknown environment using sequential queries

Data and/or labels Internal params Choose queries



Collect labels

Assume: A pool of unlabeled data

Queries: present instances and get labels (costly)

Goal: find all positive instances quickly (rewards)

Related to multi-armed bandits and Bayesian optimization

Artwork by [Javad Azimi]

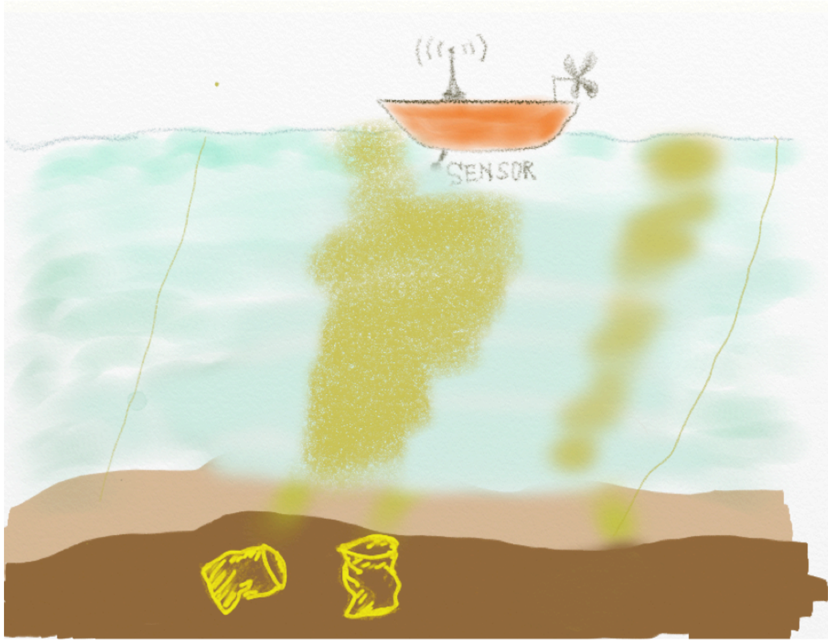
Environmental Monitoring

Search for polluted areas using a mobile sensor

Sensor measurements are costly

Decide where to collect measurements, based on:

- Previous measurements
- Spatial smoothness



Cartoon by Ying Yang

Search for Interesting Books

Graph of possible suggestions based on pairwise similarity

Find all good books books, based on

- **Previous books and impressions**
- **The graph connectivity**

[https://atechnologyjobisnoexcuse.com/2006/09/network-of-amazon-products/]

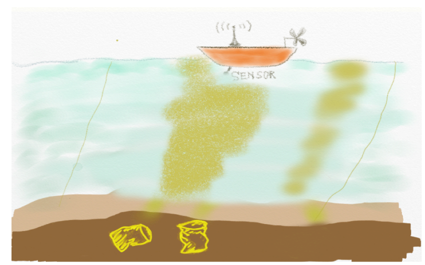
Outline / Contributions

Active search on graphs

- (NIPS 2013; UAI 2015)

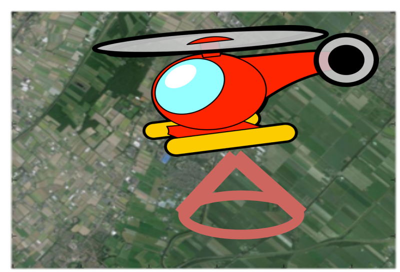
Active search with region rewards

- (AISTATS 2014;2015)



Active search with region queries

- (AAAI 2017)



Fast active search using conjugate sampling

- (in preparation)

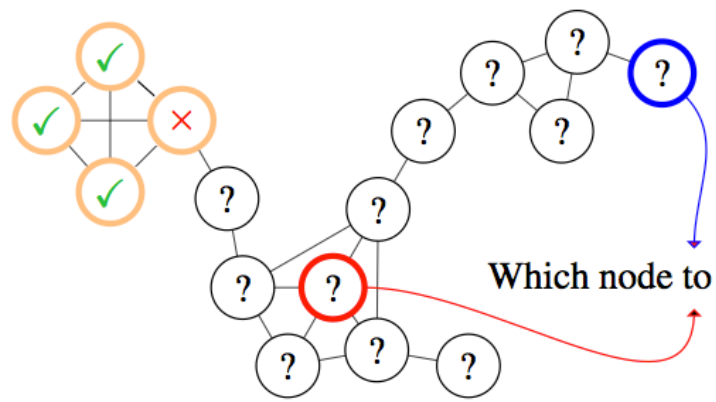
Auton Lab

Active Search on Graphs: Problem Definition

Assume: known graph; unknown labels

Task: find all  nodes using the fewest label queries

Question: which nodes to query?

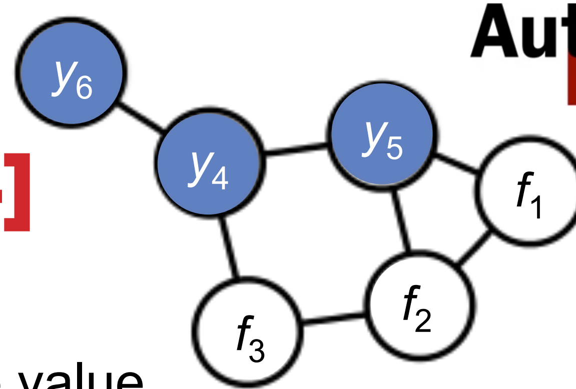


Task breakdown:

Exploration: active learning, reduce model uncertainty [NIPS 2013]

Plus Exploitation: check the likely positives, collect rewards [UAI 2015]

Gaussian Random Fields [Zhu et al., 2004]



Define f: true node value, y: observed node value,

L: Graph Laplacian = Degree – Adjacency = $\begin{pmatrix} 2 & -1 & & \ldots \\ -1 & 3 & -1 & \ldots \\ & -1 & 2 & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}$

A Bayesian model where adjacent nodes tend to have same labels

Prior

$$E(\mathbf{f}) = \frac{1}{2}\sum_{i\sim j}(f_i - f_j)^2 = \frac{1}{2}\mathbf{f}^\top L\mathbf{f}$$

$$p(\mathbf{f}) \propto \exp(-E(\mathbf{f})) \sim \mathcal{N}(0, L^{-1})$$

Relax y to real values, **Observe** y_S on set S, posterior is Gaussian with

$$\mathbb{E}(\mathbf{f} \mid \mathbf{y}_S) = \begin{cases} \mu_i = y_i, & \text{if } i \in S, \\ d_i\mu_i = \sum_{j\sim i}\mu_j, & \text{otherwise,} \end{cases} \qquad \mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S) = \begin{pmatrix} 0 & & \\ & (L_{UU})^{-1} & \\ & & 0 \end{pmatrix}$$

Good Exploration Similar to Experimental Designs

Optimal Design [Gergonne, J. D. 1815]

Design experiments to minimize some metric of model uncertainty in a look-ahead fashion

D-optimality (entropy)

V-optimality (variance)

Σ-optimality – Our contribution

[By NOAA (http://www.photolib.noaa.gov/htmls/theb1982.htm) [Public domain], via Wikimedia Commons]

Baseline 1: D-Optimality

Minimize posterior differential entropy

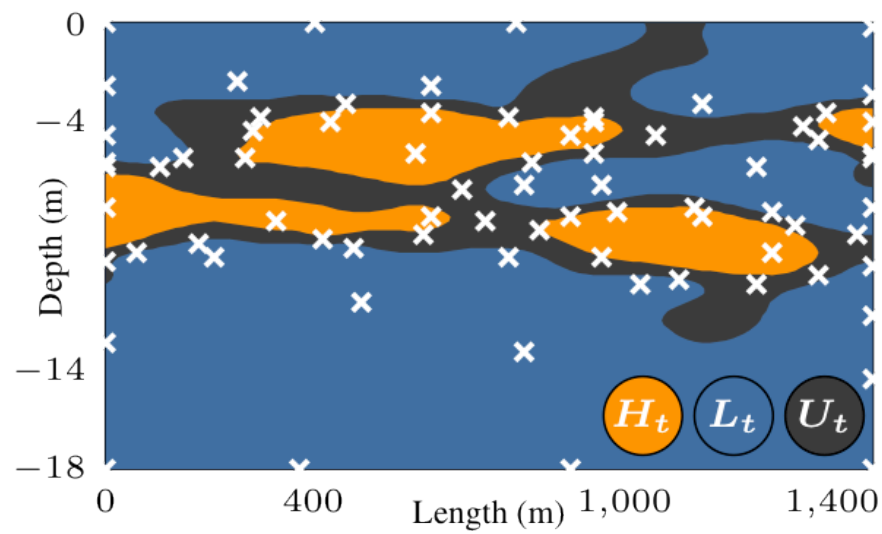
$$\min_S H(\mathbf{f} \mid \mathbf{y}_S) \simeq \log\det(\mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S))$$

Greedy application chooses by marginal variance at current step

$$\arg\min_s H(\mathbf{f} \mid \mathbf{y}_{S\cup\{s\}}) = \arg\max_s H(y_s \mid \mathbf{y}_S)$$
$$= \arg\max_s \mathrm{Var}(y_s \mid \mathbf{y}_S)$$

Not a true look-ahead measure

- Sensor placement [Krause 2008]
- GP-Bandit [Srinivas 2010]
- Level set estimation [Gotovos 2013]
- Bandits on graphs [Valko 2014]



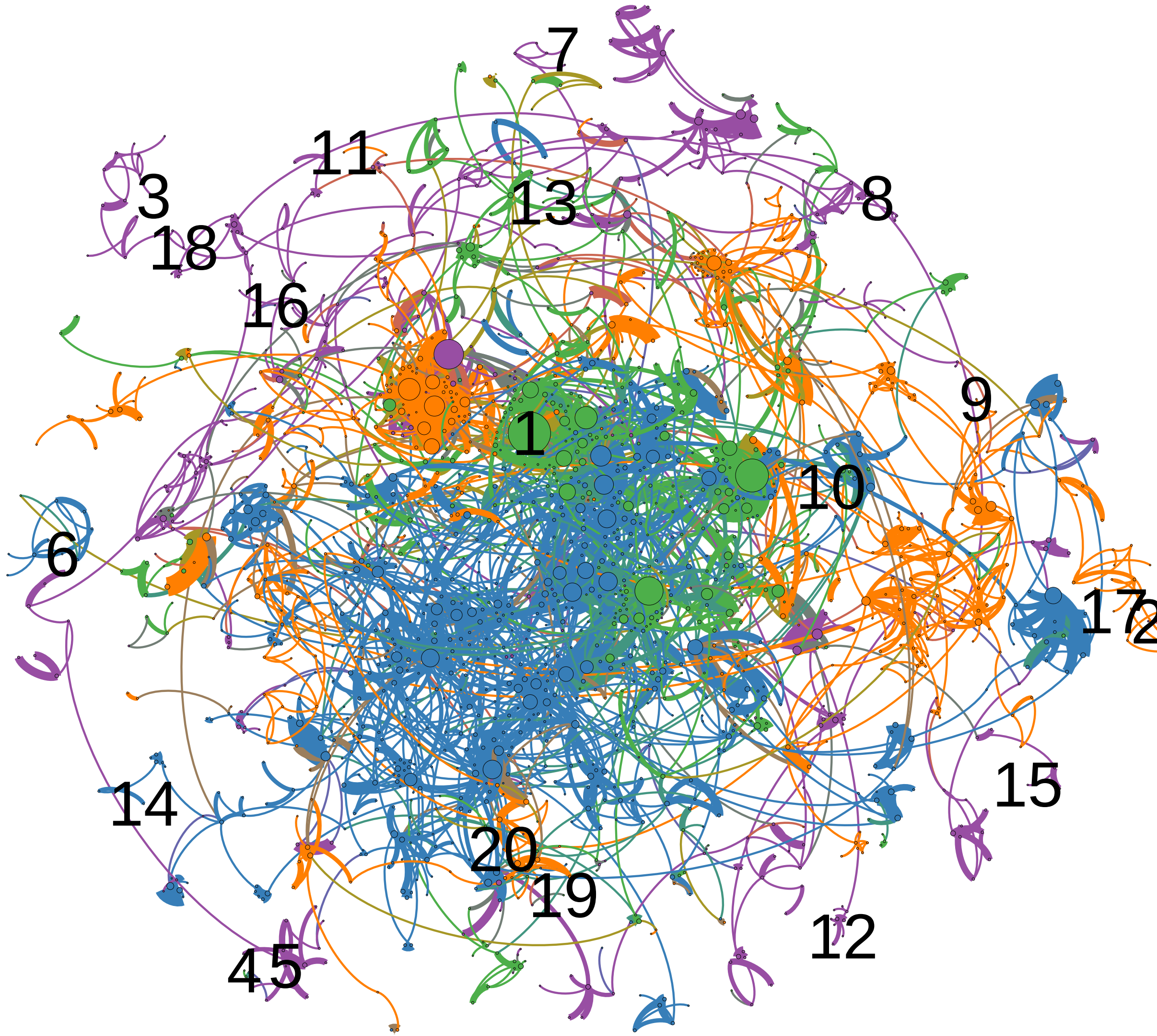
Waste samples at boundaries

Baseline 1: D-Optimality Picks Outliers

Choose the periphery

DBLP Coauthorship graph
1711 nodes, 2898 edges.
Labels (author area):

- Machine learning
- Data mining
- Information retrieval
- Database



Baseline 2: V-Optimality

True look-ahead measure

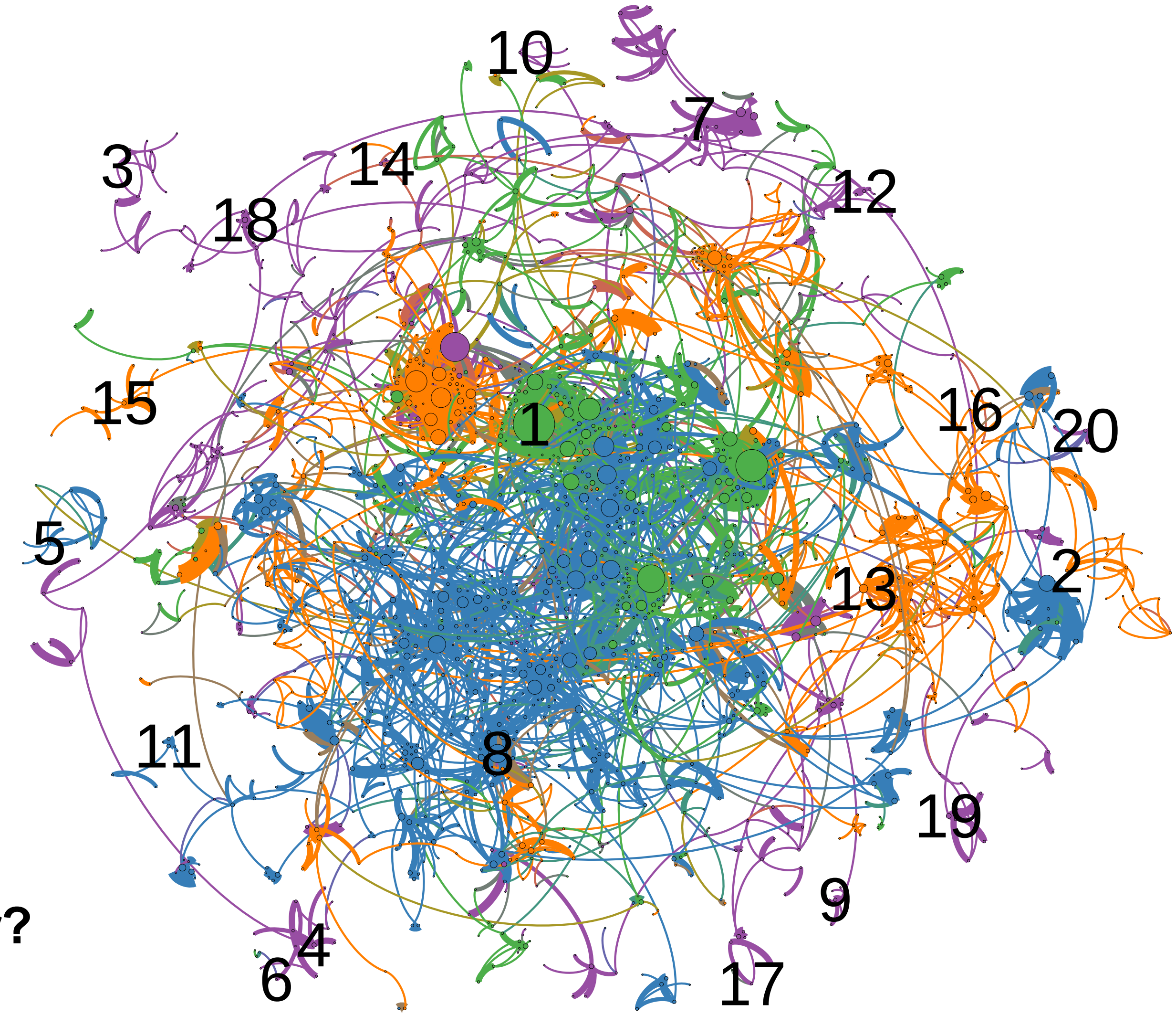
Minimize the sum of variance of the labels

$$\mathrm{loss}(\mathbf{y}, \mathbf{f}) = \sum_{1=1}^{n} (y_i - f_i)^2$$

Trace of posterior covariance matrix
[Ji & Han 2012]

$$\min_S R_V(S)$$
$$= \mathrm{tr}\left(\mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S)\right)$$

Improves
Can we do even better?



Our Approach: Σ-Optimality and Active Surveying

Bayesian optimal active search and survey [Garnett 2012]

Aims to predict the average of node values

$$\text{loss}(\mathbf{y}, \mathbf{f}) = \left(\sum_{1=1}^{n} y_i - \sum_{i=1}^{n} f_i \right)^2$$

Use GRF posterior distribution

$$(\mathbf{f} \mid \mathbf{y}_S) \sim \mathcal{N}\left(\mathbf{E}(\mathbf{f} \mid \mathbf{y}_S), \text{Cov}(\mathbf{f} \mid \mathbf{y}_S)\right)$$

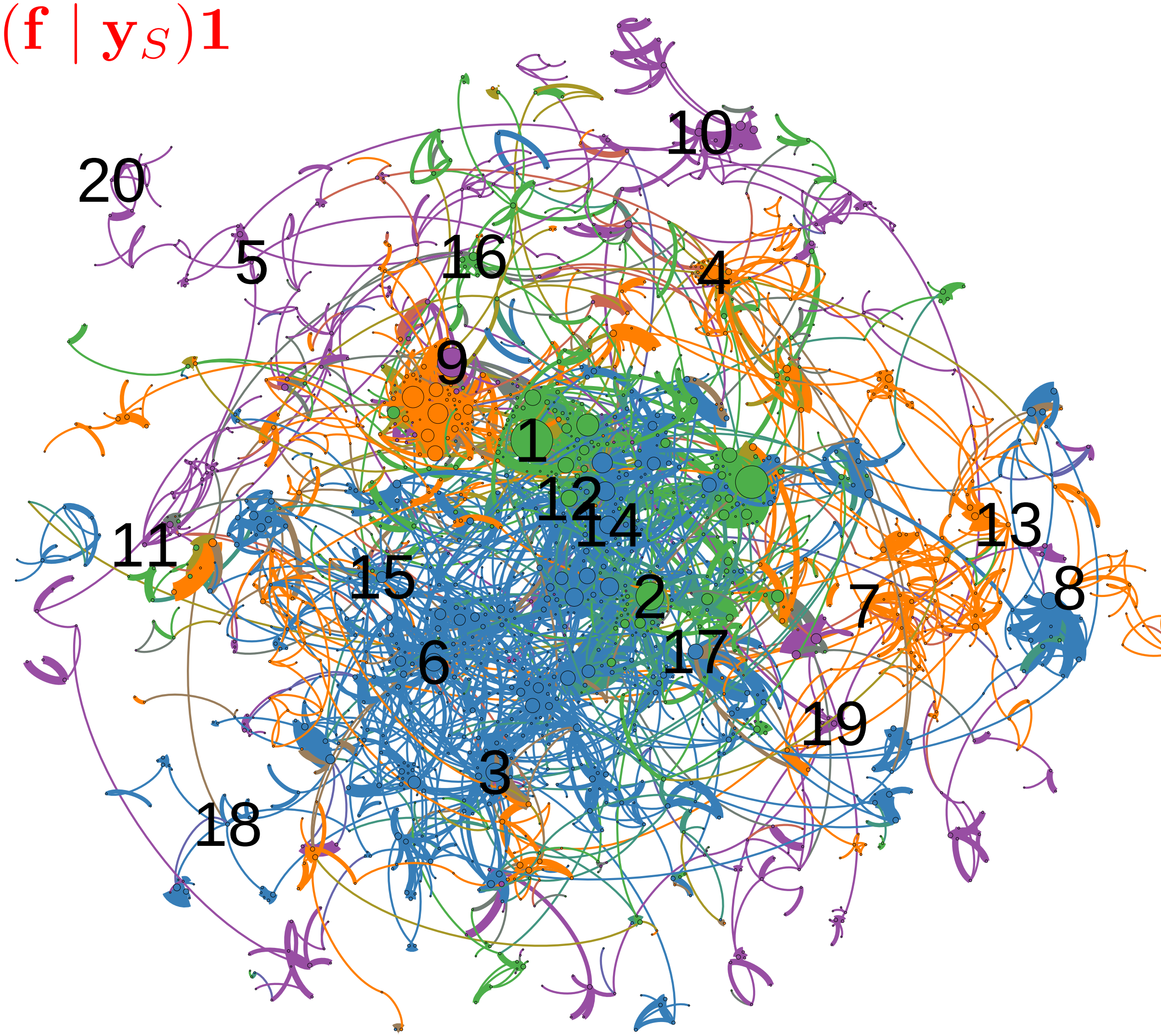
Bayesian risk minimization

$$\min_S R_\Sigma(S) = \mathbb{E}\left[\mathbb{E}\left[\text{loss}(\mathbf{y}, \mathbf{f}) \mid \mathbf{y}_S\right]\right] = \mathbf{1}^\top \text{Cov}(\mathbf{f} \mid \mathbf{y}_S)\mathbf{1}$$

Σ-Optimality on Graphs

$$\min_S R_\Sigma(S) = \mathbf{1}^\top \mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S)\mathbf{1}$$

Cluster centers!

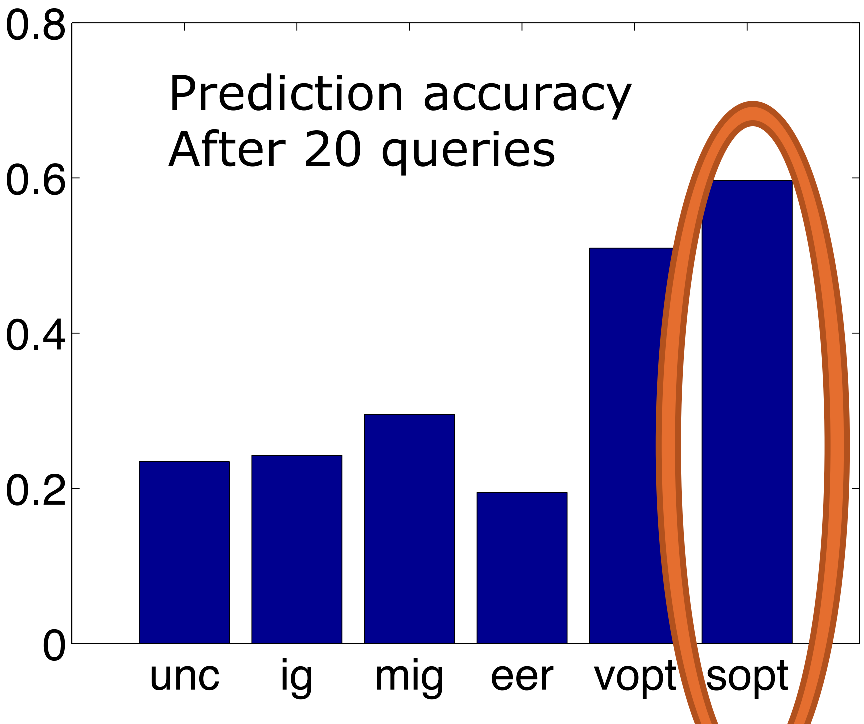


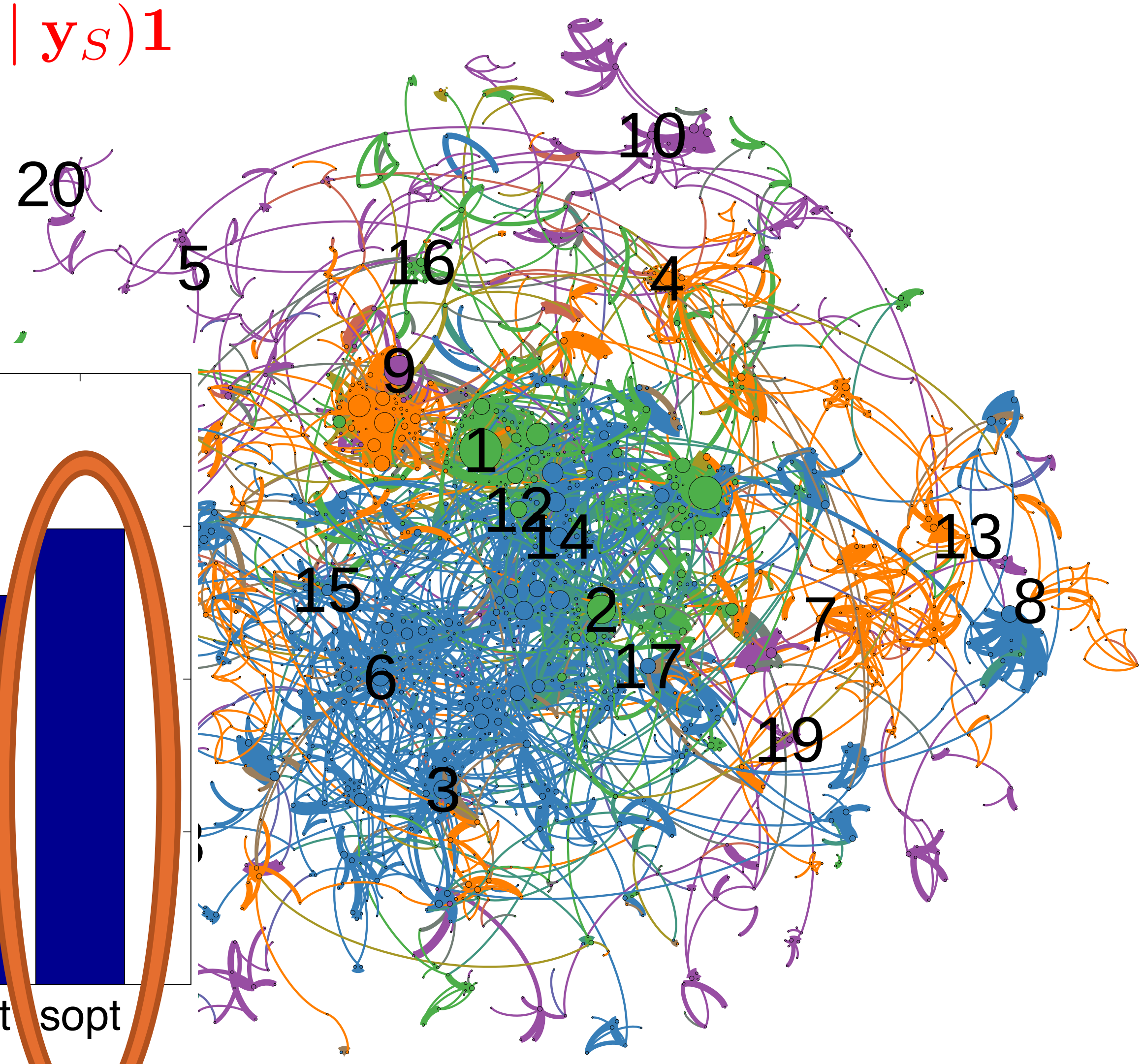
Σ-Optimality on Graphs

$$\min_{S} R_{\Sigma}(S) = \mathbf{1}^{\top}\mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S)\mathbf{1}$$

Cluster centers!

Better active learning accuracy





Insights? Break It Down to Greedy Application

Simplify notations $$\left(\mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S)\right)_{ij} = \rho_{ij}\sigma_i\sigma_j$$

ρ_{ij} posterior correlation

σ_i posterior standard deviation

Greedy selection is equivalent to

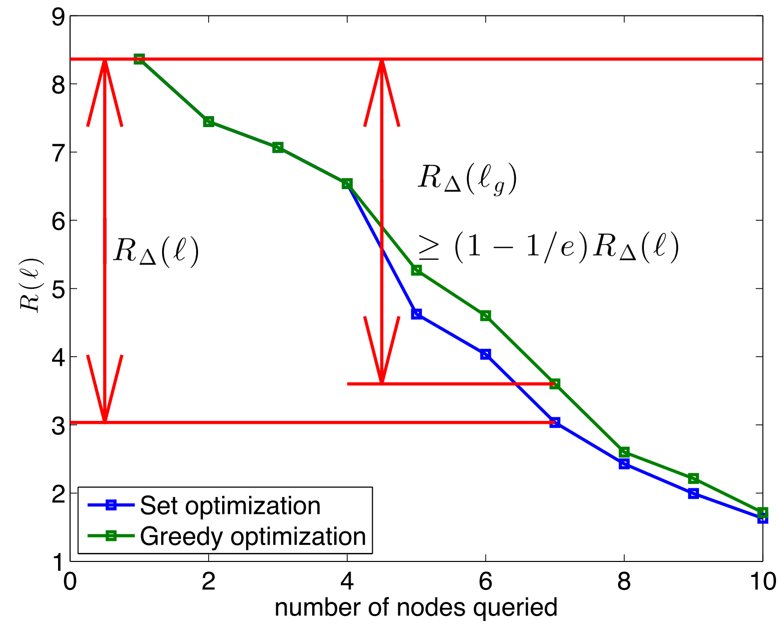
$$\text{[D-Opt Krause 2008]} \; s_{t+1} = \arg\max_i \sigma_i^2$$

$$\text{[V-Opt Ji 2012]} \; s_{t+1} = \arg\max_i \sum_j (\rho_{ij}\sigma_j)^2$$

$$\text{[}\Sigma\text{-Opt Ours]} \; s_{t+1} = \arg\max_i \sum_j \rho_{ij}\sigma_j$$

The Idea: ***L-1 more robust than L-2***

Multi-Step Look-Ahead and Greedy Selections



Greedy optimization

$$\begin{cases} s_t = \arg\min_s R(S_{t-1} \cup \{s\}) \\ S_t = S_{t-1} \cup \{s_t\} \end{cases}$$

is not equal to

Set optimization

$$\arg\min_S R(S)$$

However, we show near-optimality:

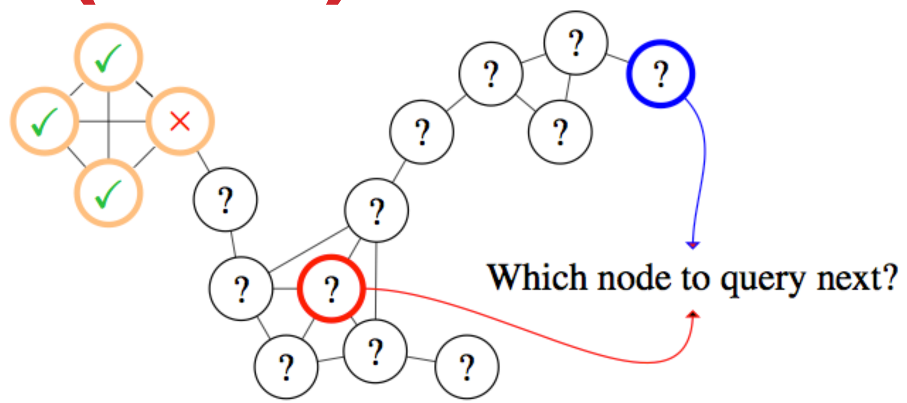
$$R(\emptyset) - R(S) \geq R(\emptyset) - R(S_t) \geq \left(1 - \frac{1}{e}\right)\left(R(\emptyset) - R(S)\right)$$

For D-, V-, Σ-optimality, due to

- **Monotone decreasing risk**
- **Diminishing returns (submodularity)**

[Ma et al., 2013]

Active Search and Upper Confidence Bound (UCB)



Choose between

Exploration: active learning, reduce model uncertainty [NIPS 2013]

Plus Exploitation: check the likely positives, collect rewards [UAI 2015]

UCB score = immediate reward + information gain (D-optimality)

$$s_{t+1} = \arg\max_i \mu_t(i) + \alpha_t \sigma_t(i)$$

$$\text{where} \begin{cases} \mu_t(i) = \mathbb{E}(f_i \mid y_{s_1}, \ldots, y_{s_t}) \\ \sigma_t(i) = \mathrm{Var}(f_i \mid y_{s_1}, \ldots, y_{s_t}) \end{cases}$$

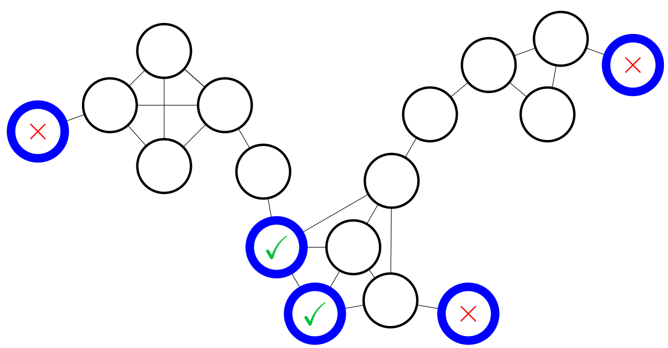
Active Search on Graphs [Ma et al., 2015]

Auton Lab

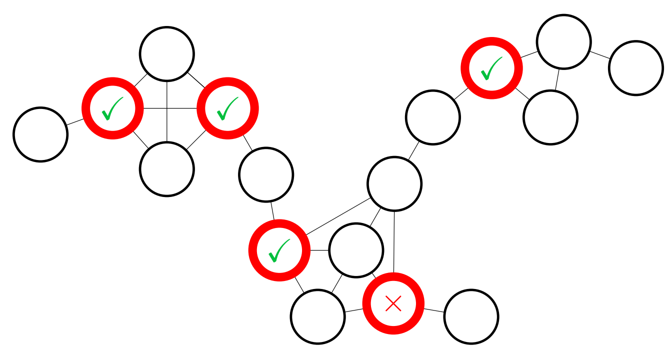
Goal: find all positive nodes

GP-SOPT $$s_{t+1} = \arg\max_i \mu_t(i) + \alpha_t \cdot g_t(i)$$

$$\text{where,} \quad g_t(i) = \sum_{j \in V} \rho_{ij} \sigma_j$$



Choices in previous work



Choices by our algorithm

Active Search on Graphs Experiments

Recall of positive nodes against the number of queries.

Experiment

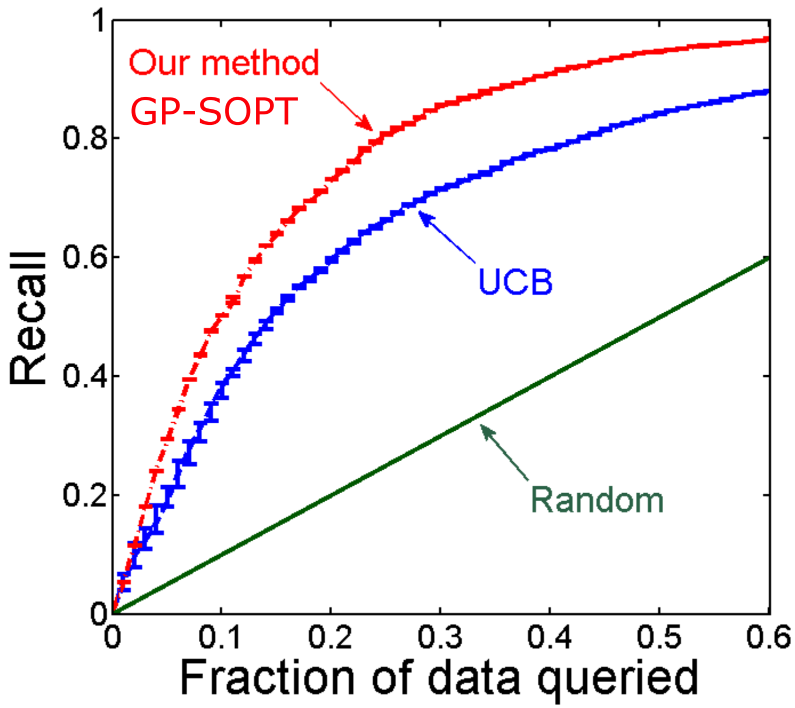
Nodes: 5000 populated places

Edges: wikipedia links

Search: 725 capitals

among countries, cities,

towns and villages



Regret Analysis

Define Regret	$R_T = \sum_{t=1}^{T} f(s_t^*) - \sum_{t=1}^{T} f(s_t)$
Define Information	$\gamma_T = \max_{\lvert S\rvert \le T} \mathcal{I}(\mathbf{y}_S; f)$
Assume	$\mathbf{f}^\top L\mathbf{f} \le B^2, \quad \text{proper} \quad \alpha_t,$ $\exists d_T^* \text{ s.t. } \gamma_T \le d_T^* \log\left(1 + \frac{T}{\sigma^2 \omega_0}\right)$
GP-SOPT [Ma et al., 2015]	$R_T \le \tilde{O}(k\sqrt{T}(B\sqrt{d_T^*} + d_T^*)), \forall T$
c.f. Spectral-UCB [Valko et al., 2014]	$R_T \le \tilde{O}(\sqrt{T}(B\sqrt{d_T^*} + d_T^*)), \forall T$

Summary: Active Search on Graphs

New exploration criterion

- Σ-Optimality, GP-SOPT
- Better empirical performance on active learning and search
- Submodularity for global optimality
- Regret Analysis

Outline / Contributions

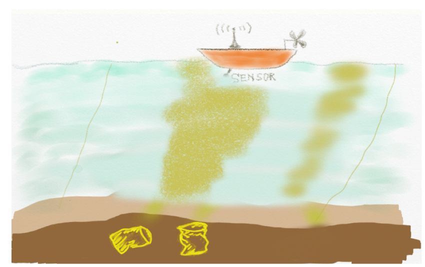
Auton Lab

Active search on graphs

- (NIPS 2013; UAI 2015)

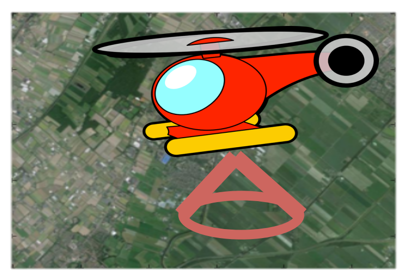
Active search with region rewards

- (AISTATS 2014;2015)



Active search with region queries

- (AAAI 2017)



Fast active search using conjugate sampling

- (in preparation)

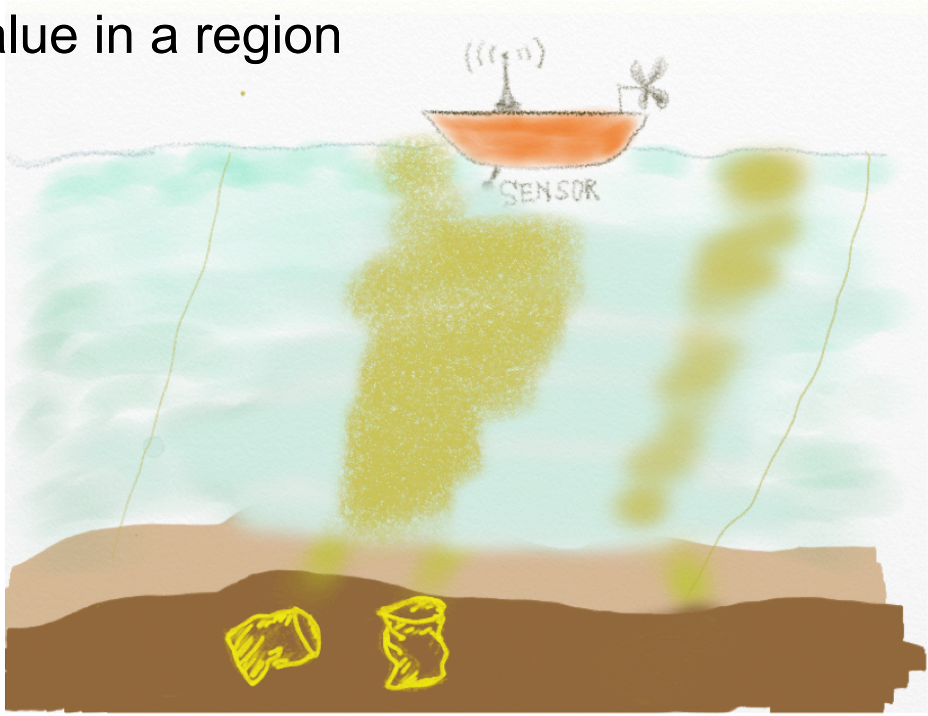
Patterns Defined by a Group of Points [Ma et al., 2014]

Search for polluted areas using a mobile sensor

Sensor measurements are costly

Find entire regions

- Reward defined by the average value in a region



Cartoon by Ying Yang

Simple Pattern: Region Integral

Assume a smooth function *f(x)*

Point observations

Choose point x_i

Observe value $z_i = f(x_i) + \varepsilon$

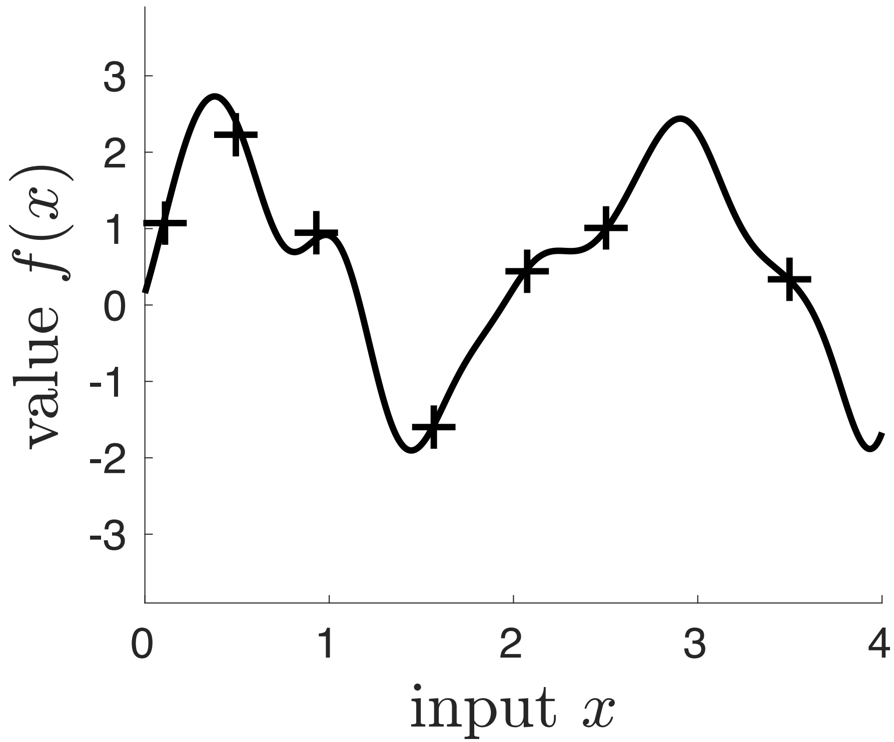
Region pattern

Pre-define regions $A_1,\ldots,A_K$.

Pattern:

region integral > threshold b

$$h_A(f) = 1_{\{\frac{1}{|A|}\int_A f \, \mathrm{d}x > b\}}$$



Simple Pattern: Region Integral

Assume a smooth function *f(x)*

Point observations

Choose point x_i

Observe value $z_i = f(x_i) + \varepsilon$

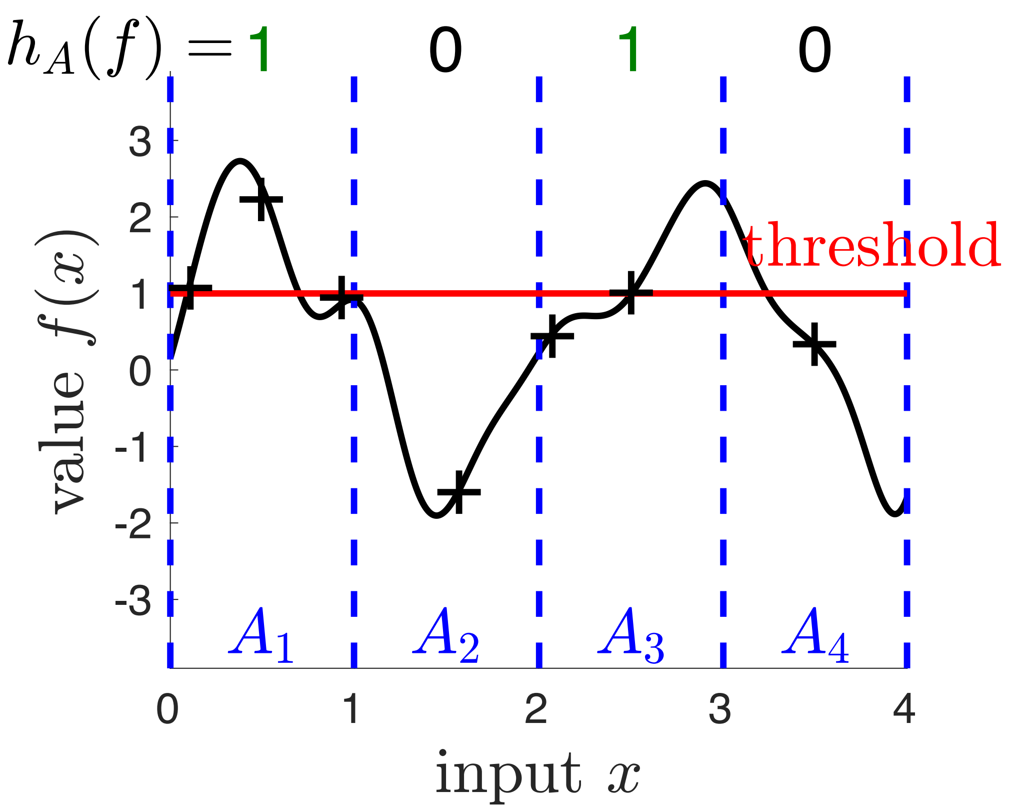
Region pattern

Pre-define regions $A_1,\ldots,A_K$.

Pattern:

region integral > threshold b

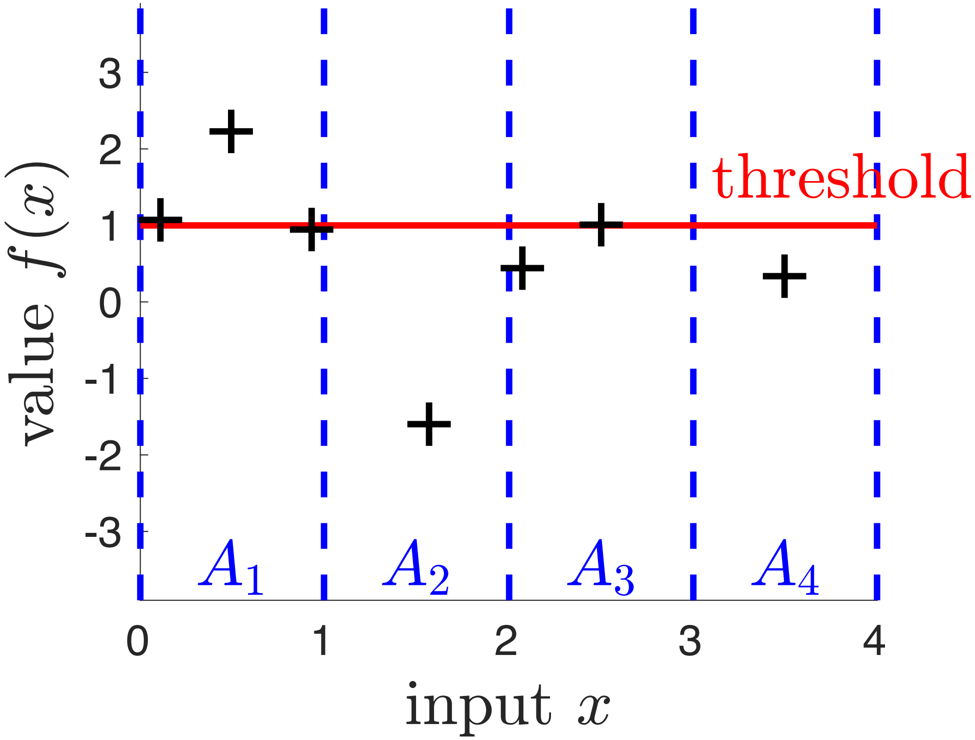
$$h_A(f) = 1_{\{\frac{1}{|A|}\int_A f\,\mathrm{d}x > b\}}$$



Infer Region Patterns Without Full Observations

However, we can only collect a few data points ("+")

- True region average requires all data points



Infer Region Patterns Without Full Observations

However, we can only collect a few data points ("+")

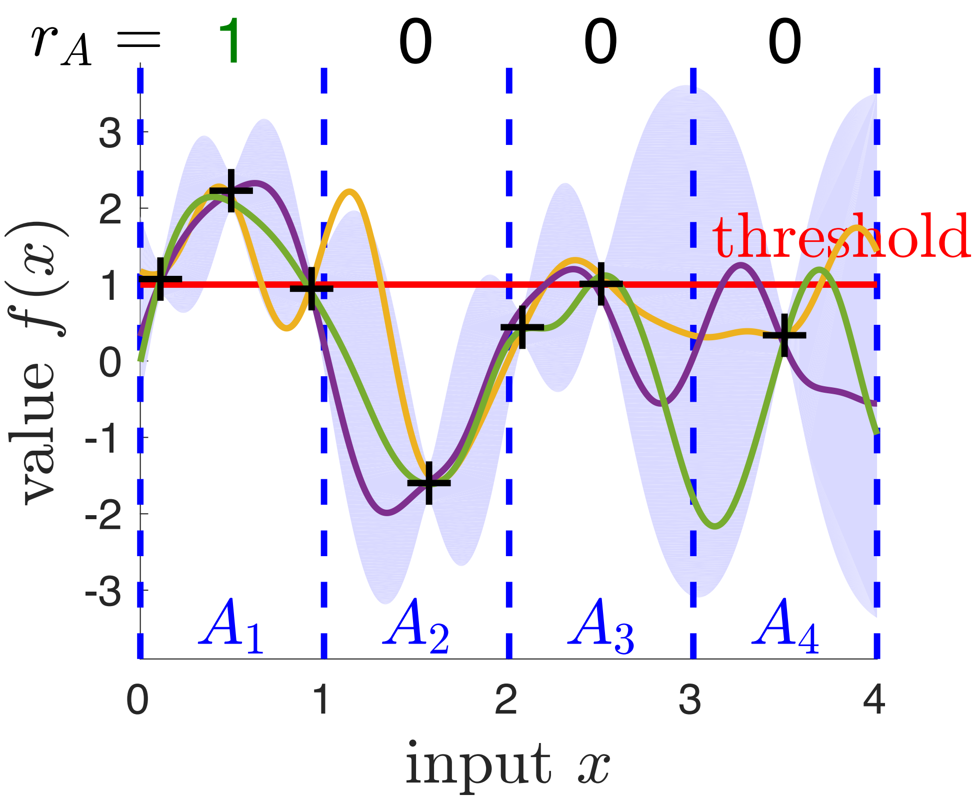
- True region average requires all data points

Instead, assume *f(x)* is drawn from a Gaussian Process (GP)

- Distribution over smooth functions
- Post. dist. given observed data

Assign rewards to a region if region integral has at least θ prob. to be greater than the threshold

$$r_A(X, \mathbf{z}) = 1_{\left\{\mathbb{E}\left(h_A(f)|X,\mathbf{z}\right)>\theta\right\}}$$



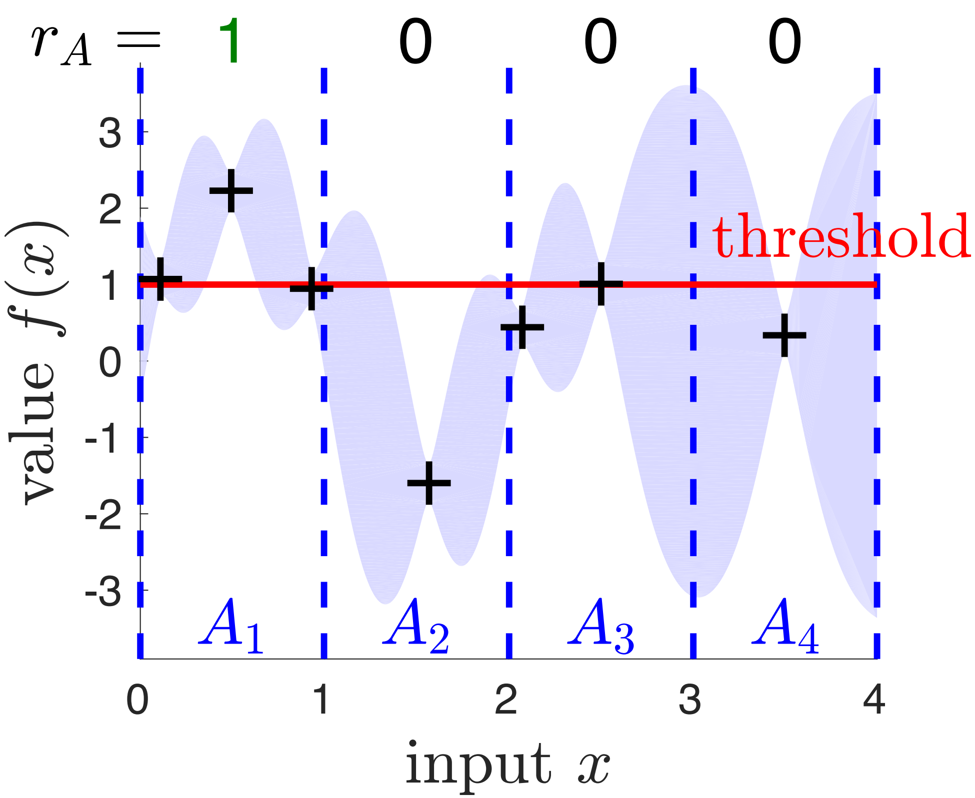
Algorithm: Maximizes Expected Reward

Reward: $r_A(X, \mathbf{z}) = 1_{\left\{\mathbb{E}\left(h_A(f)|X,\mathbf{z}\right)>\theta\right\}}$

At step t+1,
choose location x_{t+1} with maximum Monte-Carlo look-ahead estimate:

$$u(x_{t+1}) = \mathbb{E}^{z_{t+1}} \sum_k \left[r_{A_k}(X_{1:t+1}, \mathbf{z}_{1:t+1}) \right]$$

$$\text{where} \quad z_{t+1} \sim \mathcal{GP}\left(z(x_{t+1}) \mid X_{1:t}, \mathbf{z}_{1:t} \right)$$



Algorithm: Maximizes Expected Reward

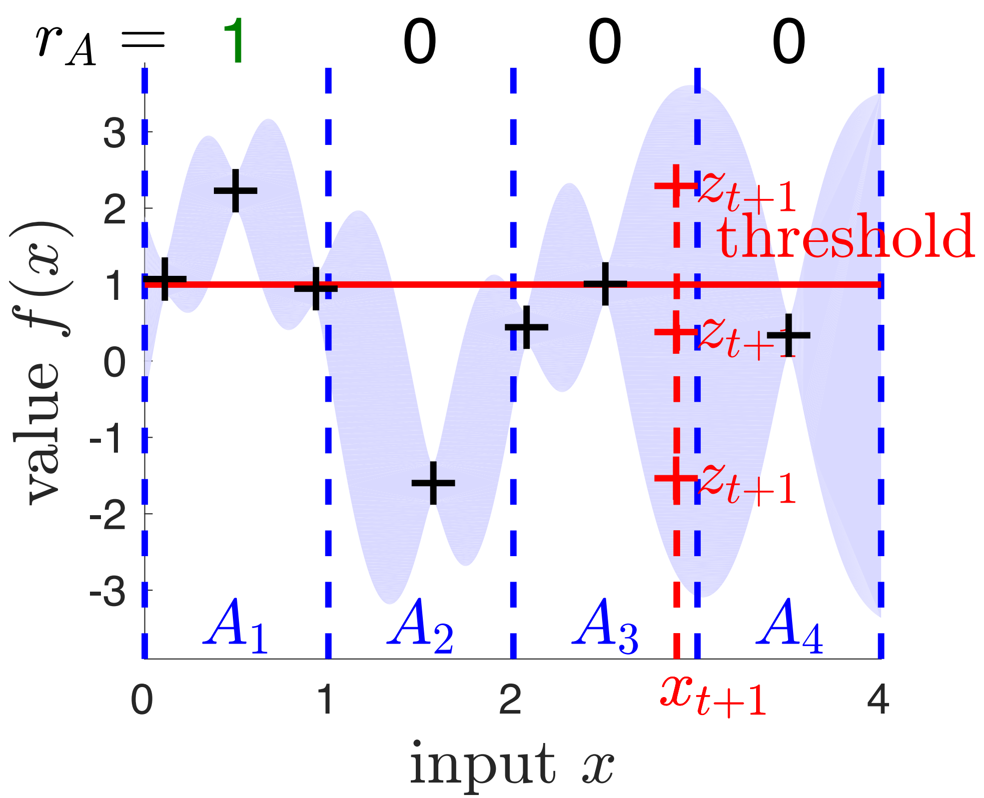
Reward: $r_A(X, \mathbf{z}) = 1_{\left\{\mathbb{E}\left(h_A(f)|X,\mathbf{z}\right)>\theta\right\}}$

At step *t*+1,
choose location x_{t+1} with maximum
Monte-Carlo look-ahead estimate:

$$u(x_{t+1}) = \mathbb{E}^{z_{t+1}} \sum_k \left[r_{A_k}(X_{1:t+1}, \mathbf{z}_{1:t+1}) \right]$$

$$\text{where } z_{t+1} \sim \mathcal{GP}\left(z(x_{t+1}) \mid X_{1:t}, \mathbf{z}_{1:t} \right)$$

Sample outcomes z_{t+1}



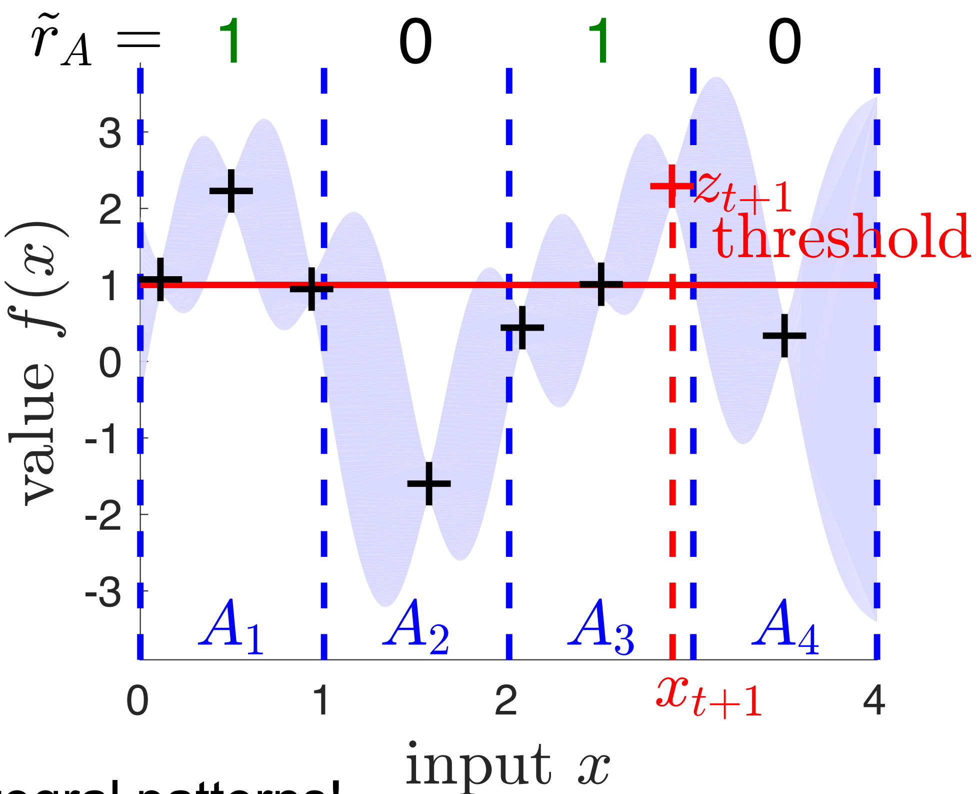
Algorithm: Maximizes Expected Reward

Reward: $$r_A(X,\mathbf{z}) = 1_{\left\{\mathbb{E}\left(h_A(f)|X,\mathbf{z}\right)>\theta\right\}}$$

At step *t*+1,
choose location x_{t+1} with maximum Monte-Carlo look-ahead estimate:

$$u(x_{t+1}) = \mathbb{E}^{z_{t+1}} \sum_k \left[r_{A_k}(X_{1:t+1}, \mathbf{z}_{1:t+1}) \right]$$

$$\text{where } \; z_{t+1} \sim \mathcal{GP}\left(z(x_{t+1}) \mid X_{1:t}, \mathbf{z}_{1:t} \right)$$



Sample outcomes z_{t+1}

Compute look-ahead reward

Average for expected rewards

Closed-form solutions for region integral patterns!

Closed-Form Solution Intuitions

For patterns defined by region integrals

If regions are well-separated (assume each query only affects one region)

Then closed-form solution reduces to

1. For each region, choose a point to reduce variance of the integral

Bayesian quadrature [Minka 2000]

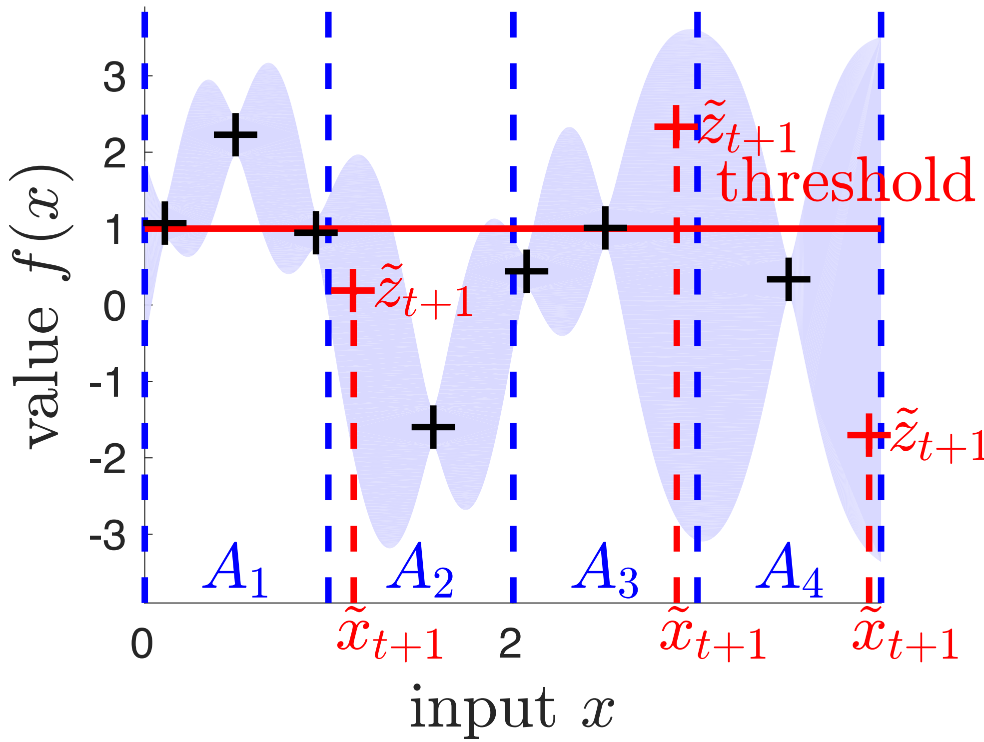
Σ-optimality

2. Compare regions UCB-style

High posterior mean

and

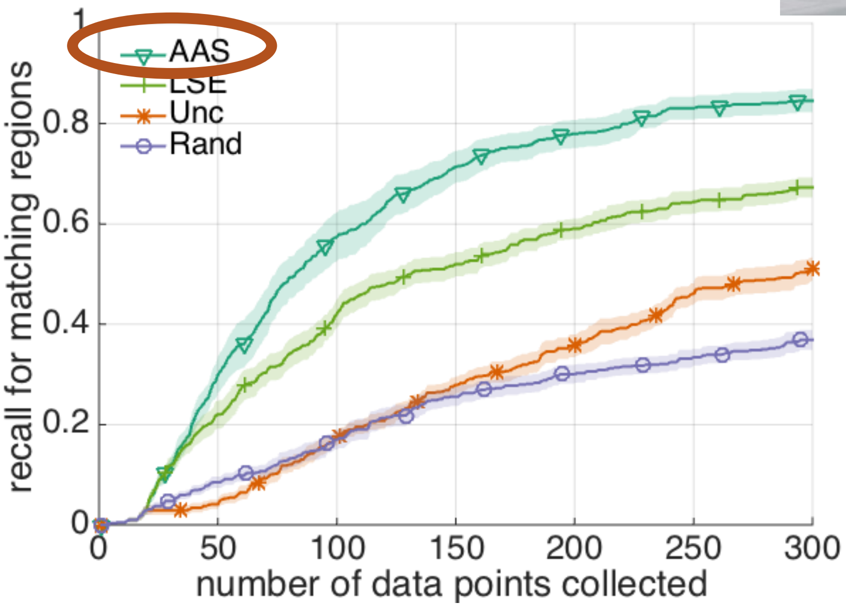
Large variance reduction

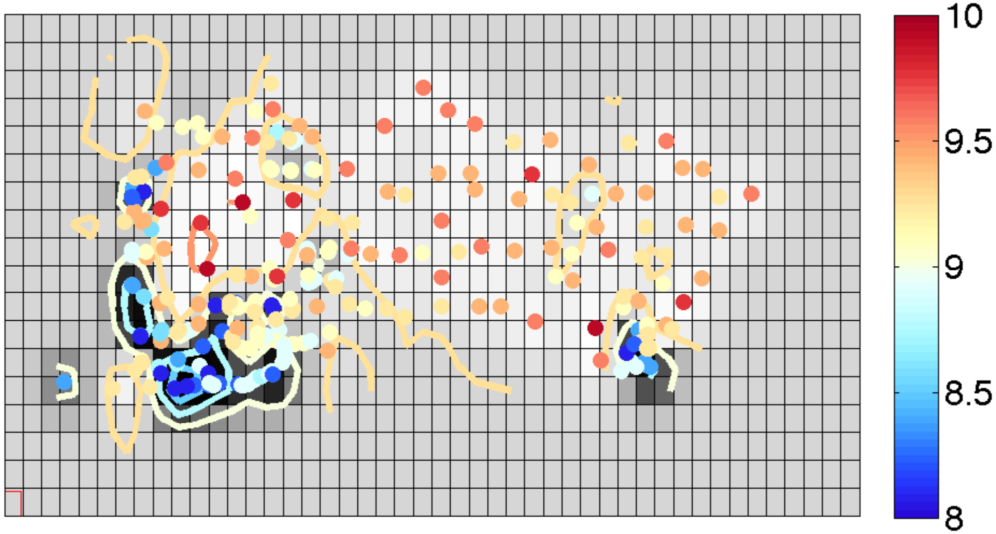


Water Quality (Dissolved Oxygen)

Recall of target regions

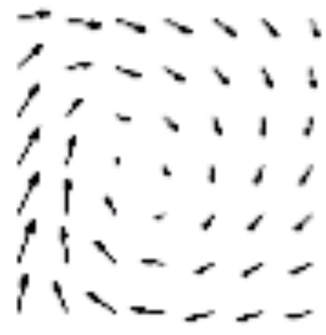
Re-picked measurements



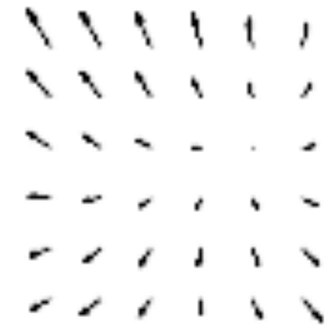


Identify Fluid Flow Vortices [Ma&Sutherland et al., 2015]

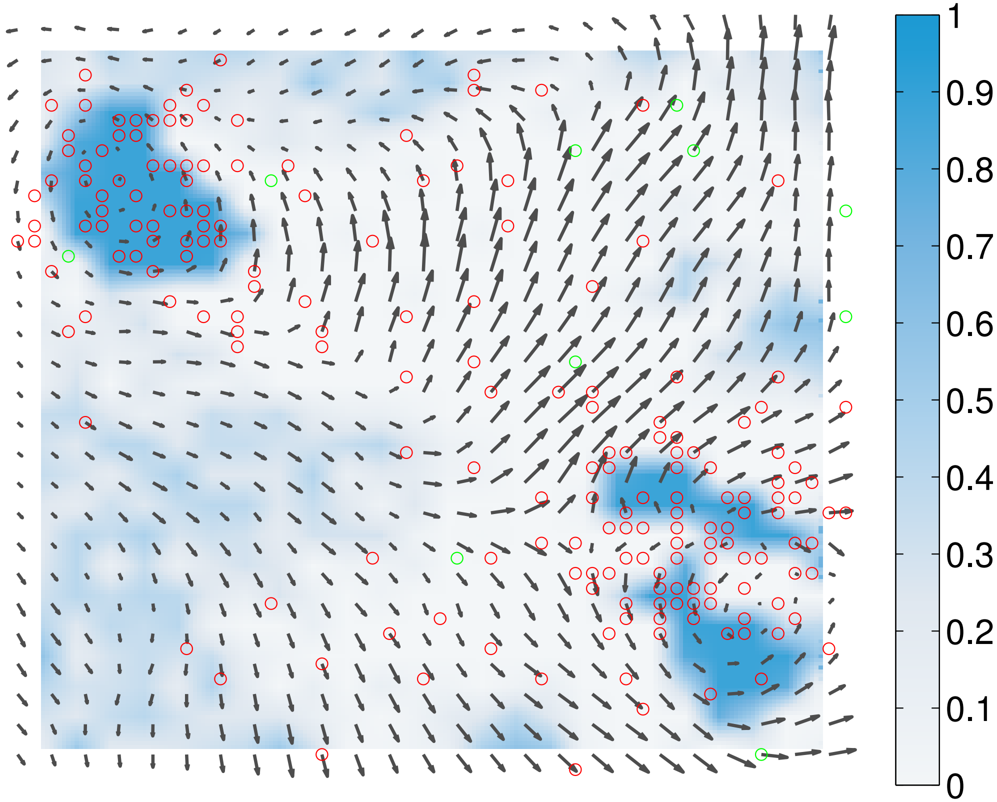
Observe point vectors
Objective overlapping windows of 11x11 that contain a vortex
Classifier 2-layer neural net

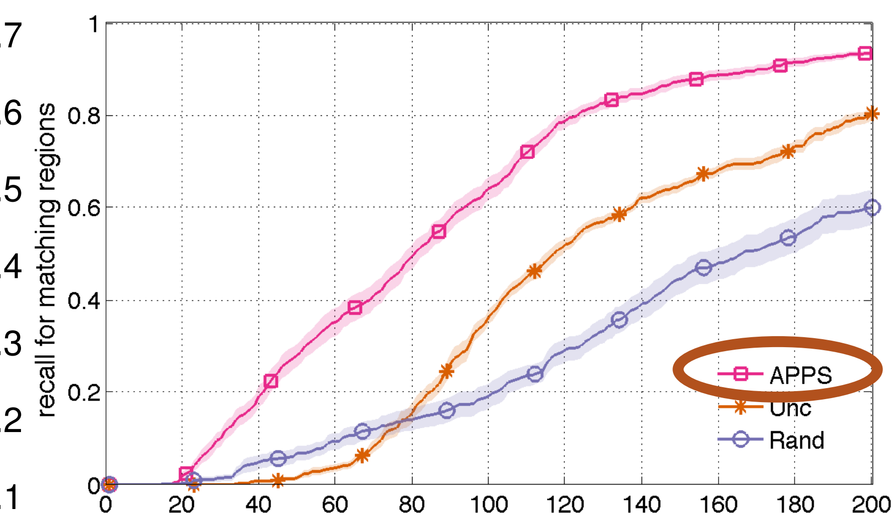


Positive example



Negative example





Summary: Active Search for Region Patterns

Bayesian expected rewards maximization

Closed-form solution has two steps

- In each region, choose a point by Bayesian quadrature
- Choose the final query by comparing regions UCB-style

Monte-Carlo approach allows for experiments with complex patterns

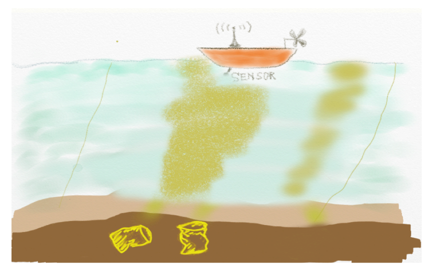
Outline / Contributions

Active search on graphs

- (NIPS 2013; UAI 2015)

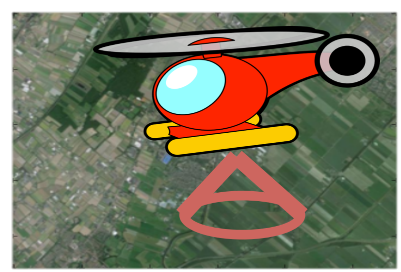
Active search with region rewards

- (AISTATS 2014;2015)



Active search with region queries

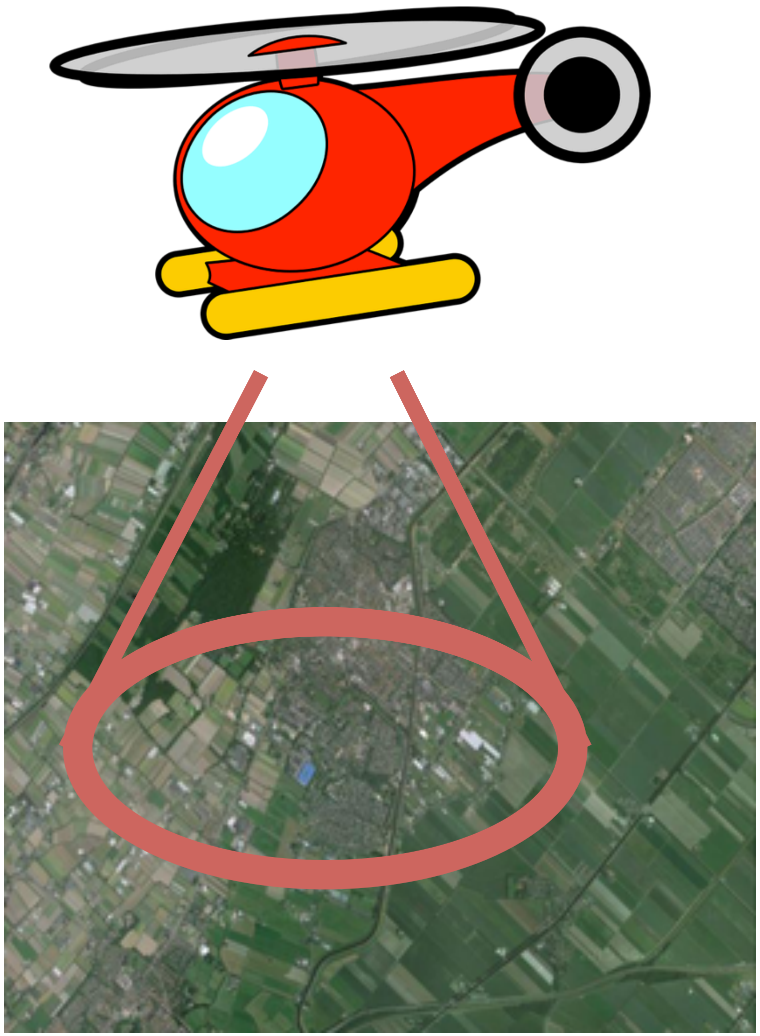
- (AAAI 2017)



Fast active search using conjugate sampling

- (in preparation)

Sparse Rewards and Region Sensing



Region sensing (aggregate value)

Task: localize the sources

Control: both altitude and position

- Radiation
- Gas leaks
- Survivors

Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

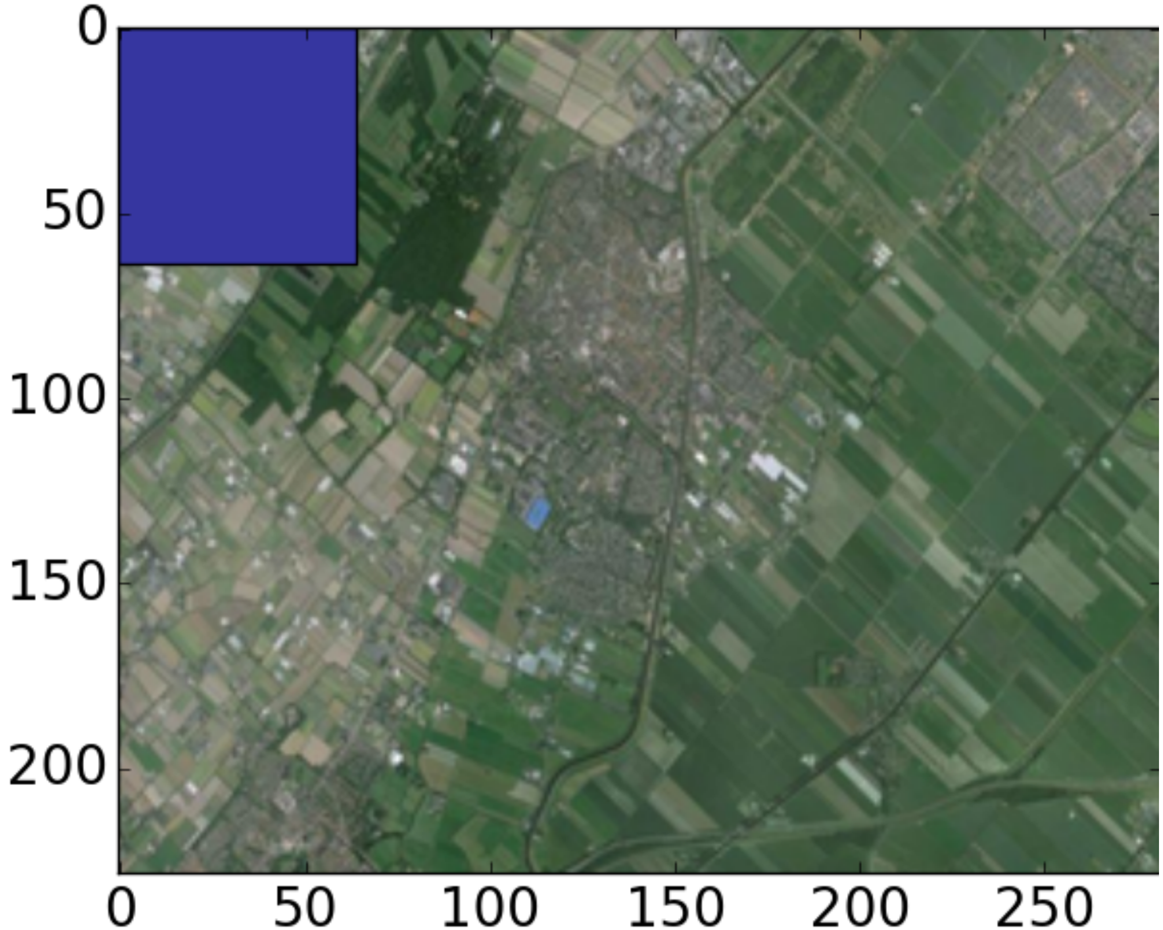
(Used a blue filter on the RGB values, yielding scalar outcomes)

Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

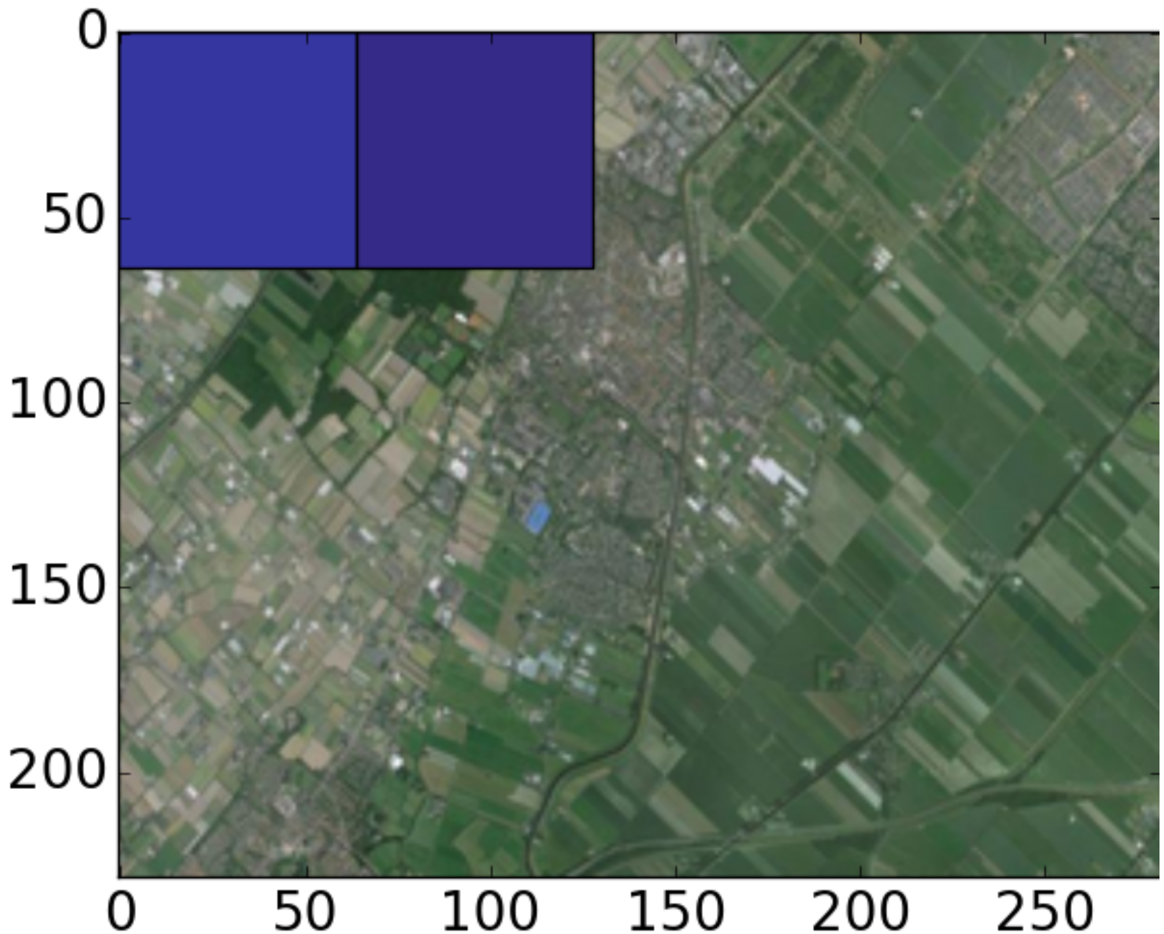


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

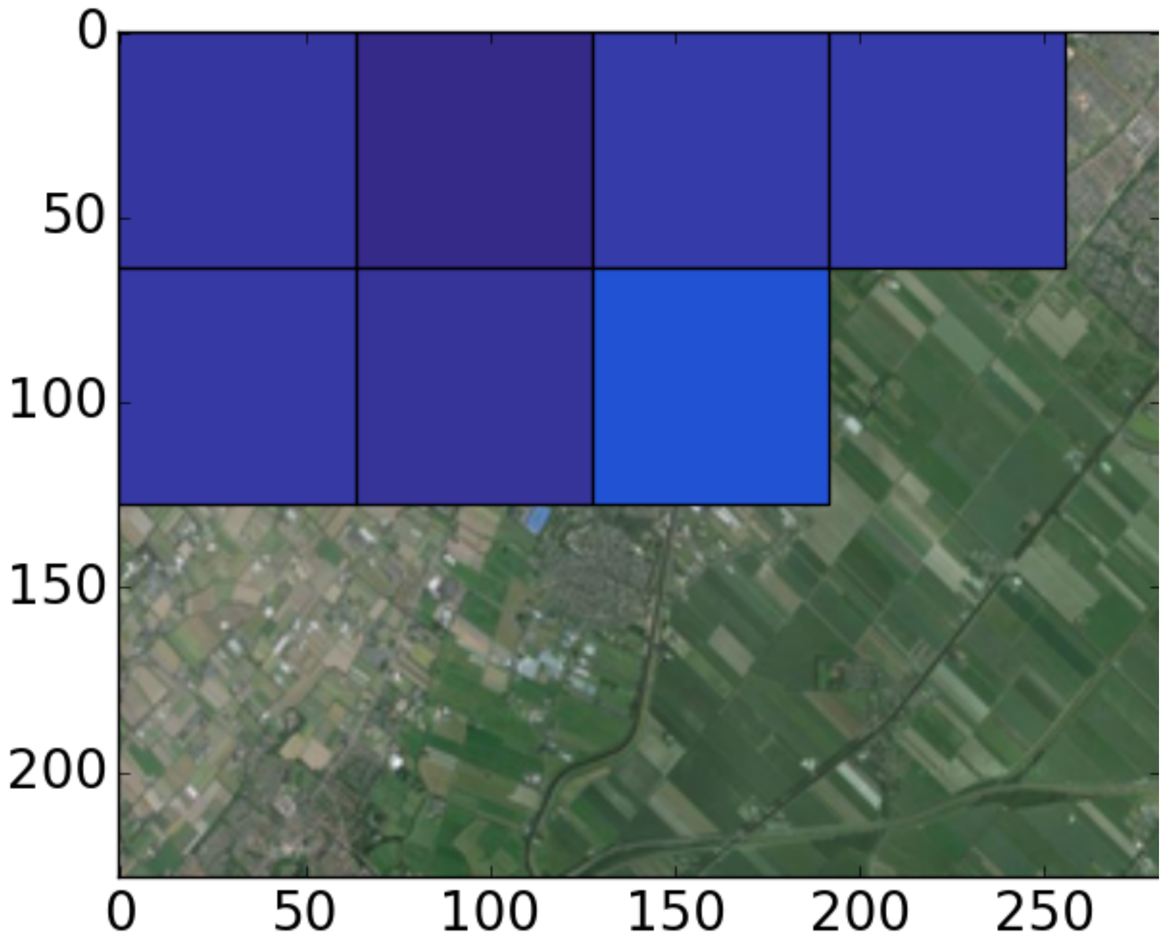


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

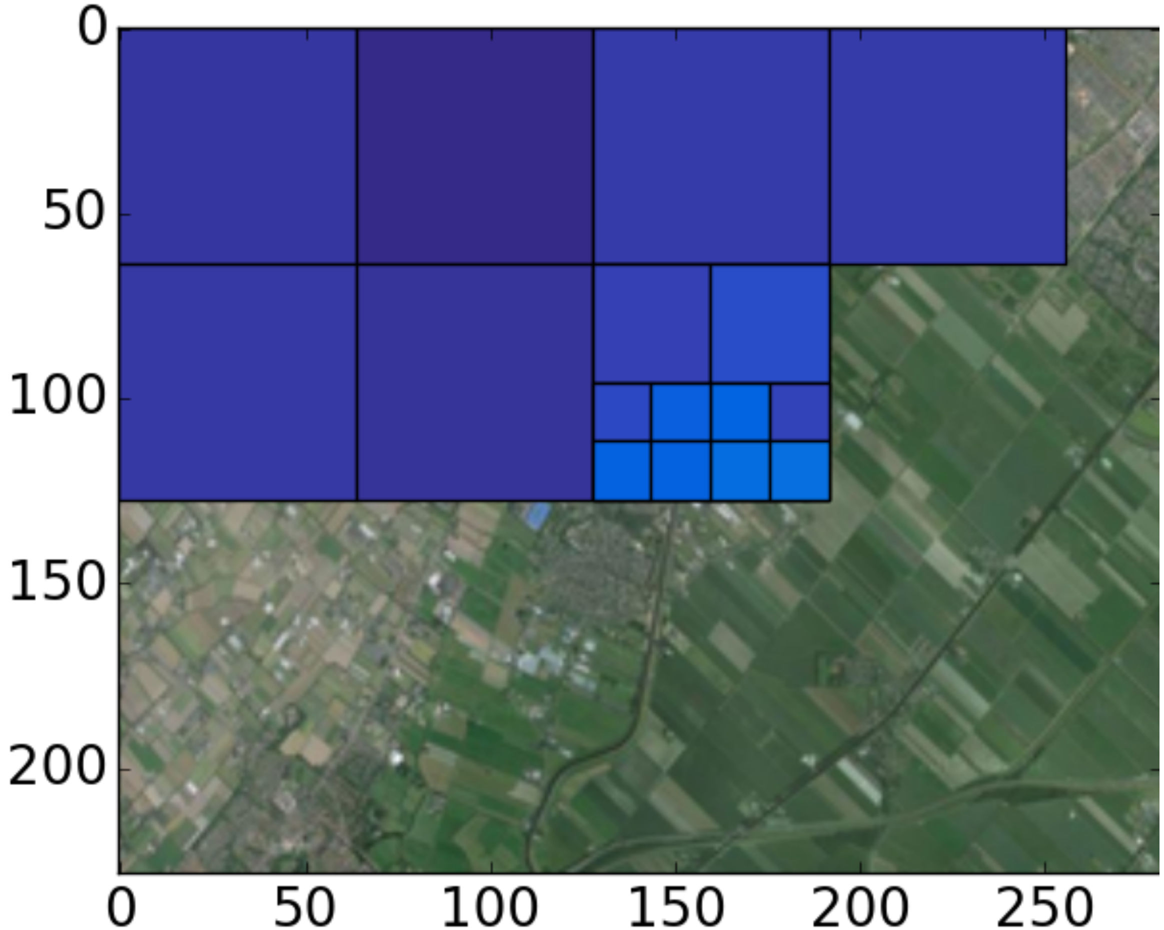


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

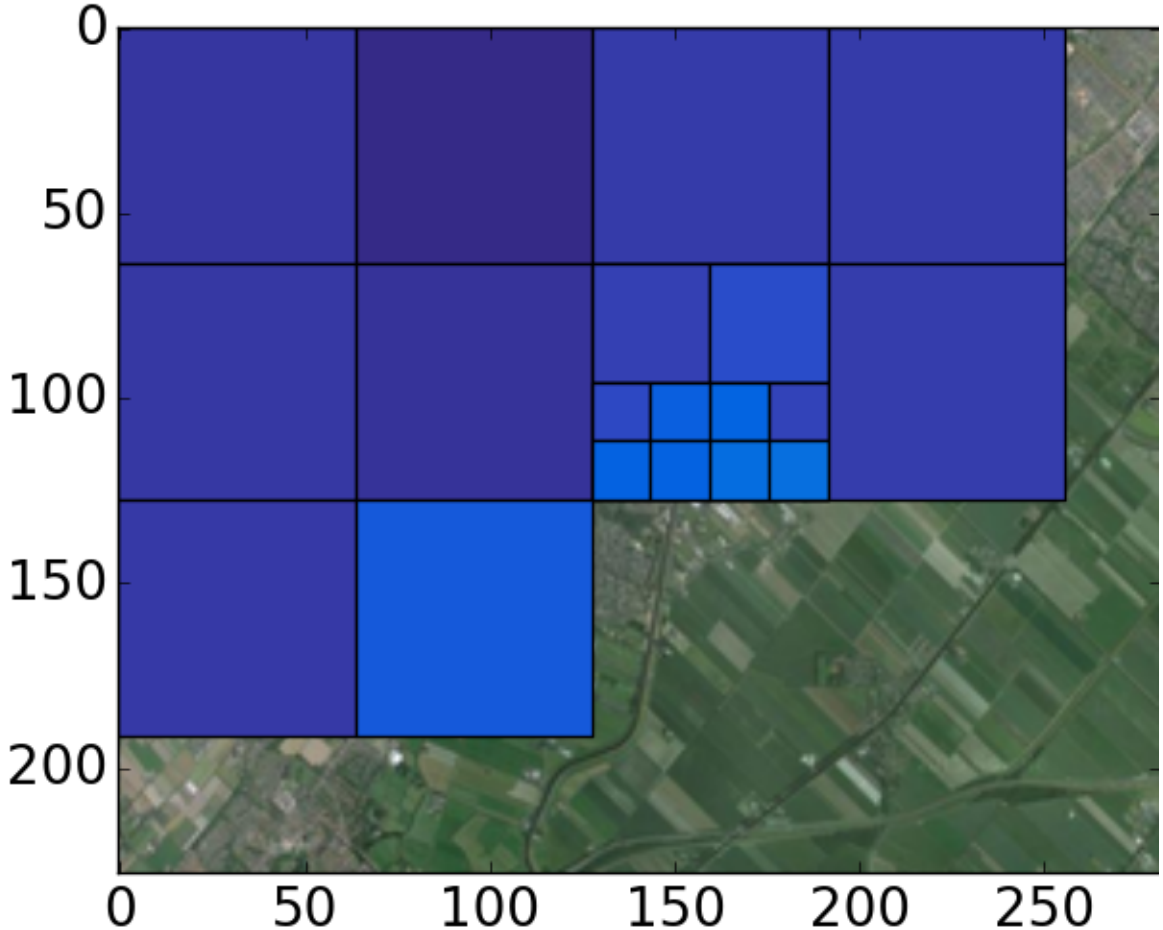


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

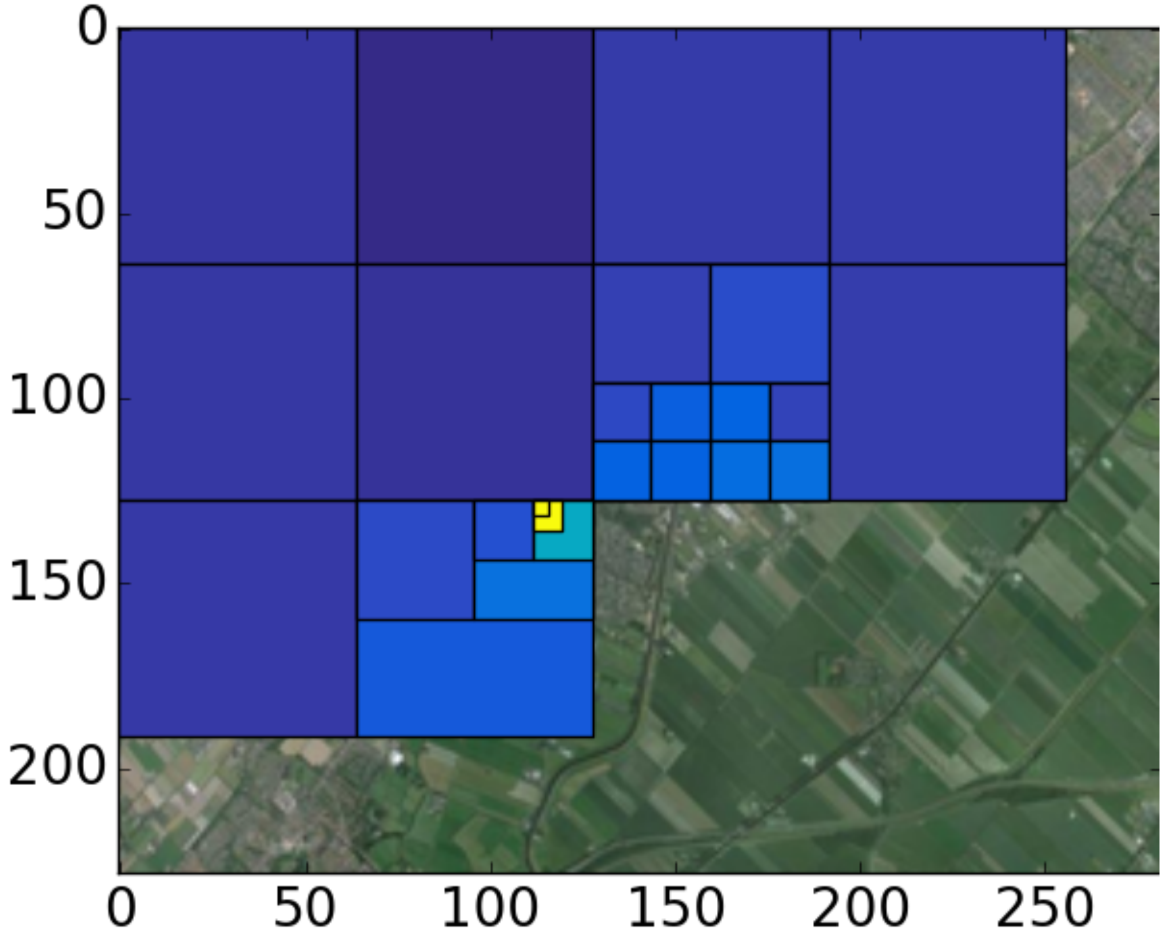


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

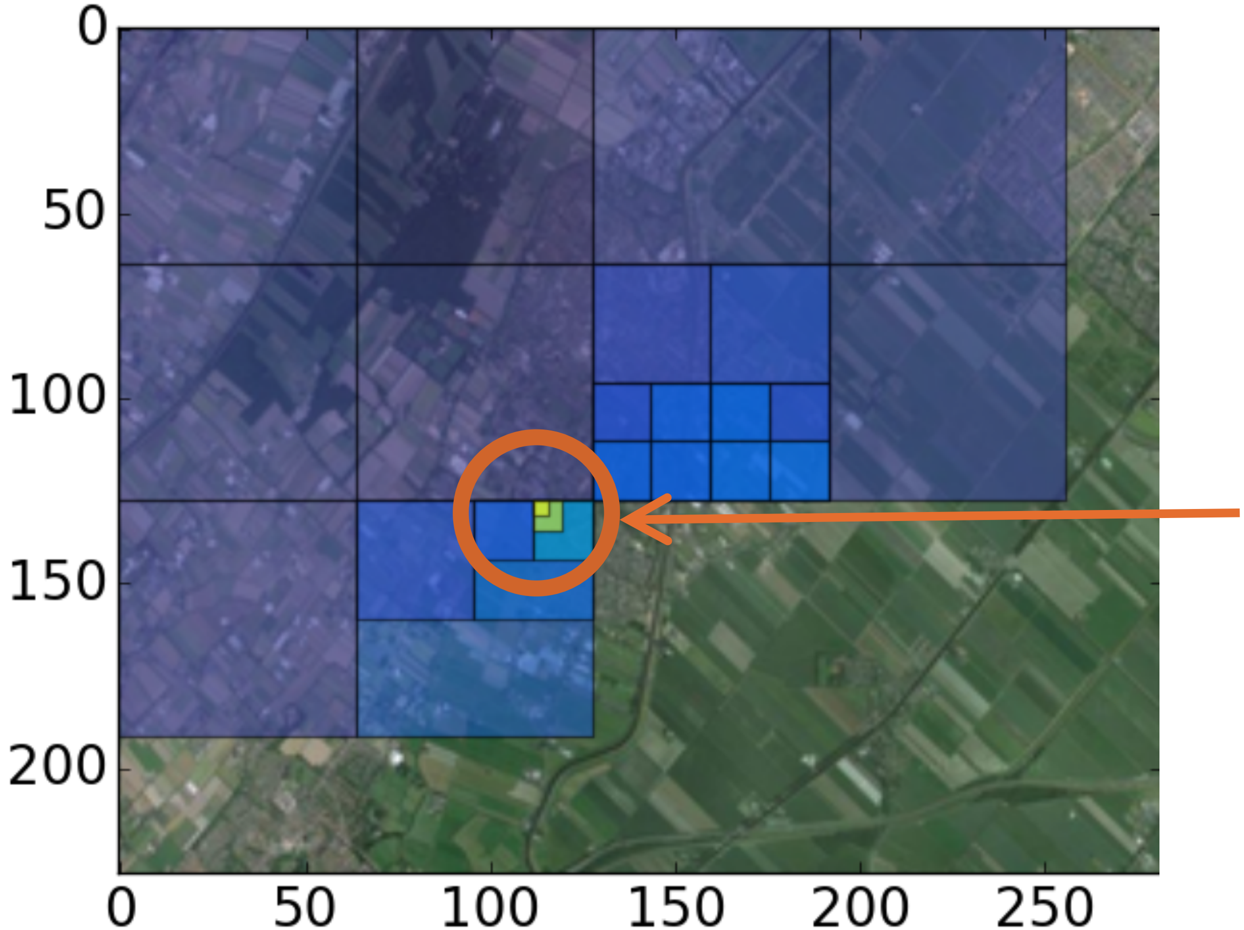


Demo Active Search

Find blue colors on a real satellite image

Simulate search and rescue in open areas

(Used a blue filter on the RGB values, yielding scalar outcomes)

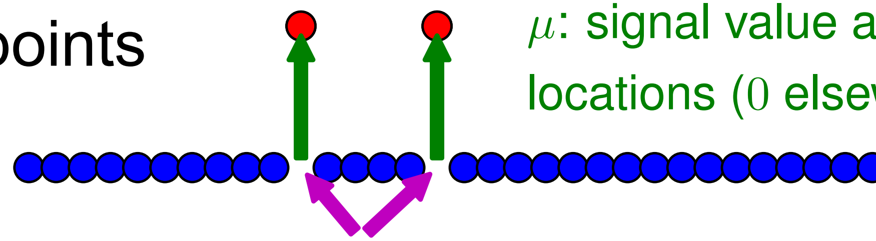


Problem Formulation

1d search, n total grid points

k-sparse signal

$\boldsymbol{\beta}^* \in \mathbb{R}_+^n$

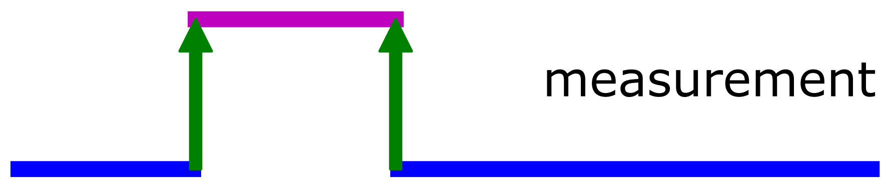


μ: signal value at nonzero locations (0 elsewhere)

S^*: set of nonzero signal locations

aggregate measurement

$\mathbf{x}_t \in \mathbb{R}_+^n, \ \|\mathbf{x}_t\|_2 = 1$



measurement strength

Sensing model $y_t = \mathbf{x}_t^\top \boldsymbol{\beta}^* + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0, 1)$

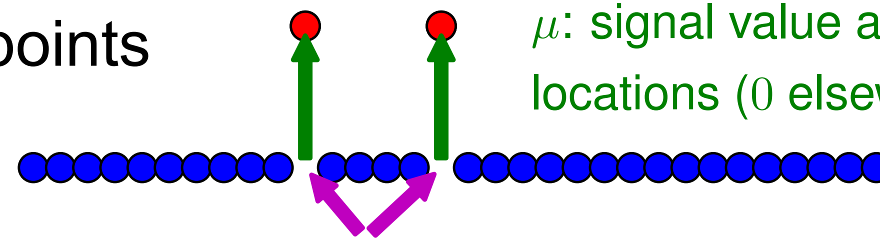
Objective: design x_t to recovery the support of β^*

Problem Formulation

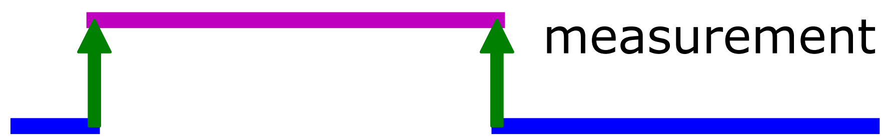
1d search, n total grid points

k-sparse signal

$\boldsymbol{\beta}^* \in \mathbb{R}_+^n$



aggregate measurement



$\mathbf{x}_t \in \mathbb{R}_+^n, \ \|\mathbf{x}_t\|_2 = 1$

Sensing model

$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta}^* + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0, 1)$$

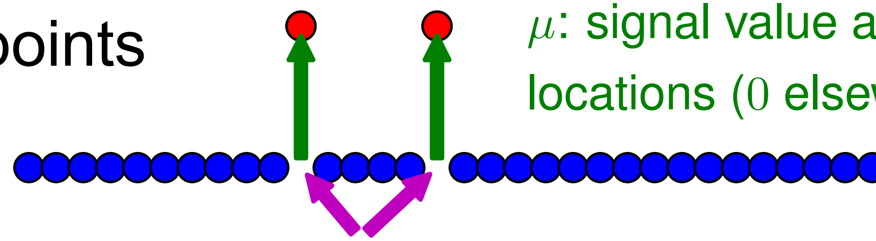
Objective: design x_t to recovery the support of β^*

Problem Formulation

1d search, n total grid points

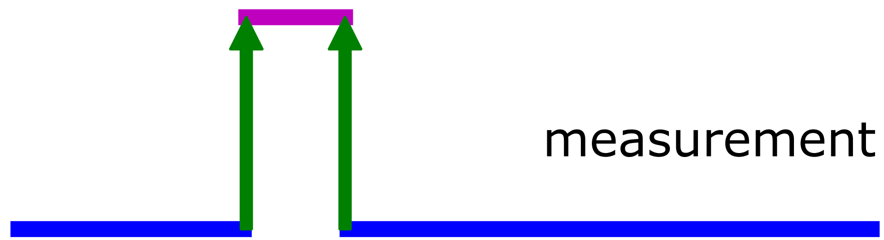
k-sparse signal

$$\boldsymbol{\beta}^* \in \mathbb{R}_+^n$$



aggregate measurement

$$\mathbf{x}_t \in \mathbb{R}_+^n, \; \|\mathbf{x}_t\|_2 = 1$$



Sensing model

$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta}^* + \varepsilon_t, \; \varepsilon_t \sim \mathcal{N}(0,1)$$

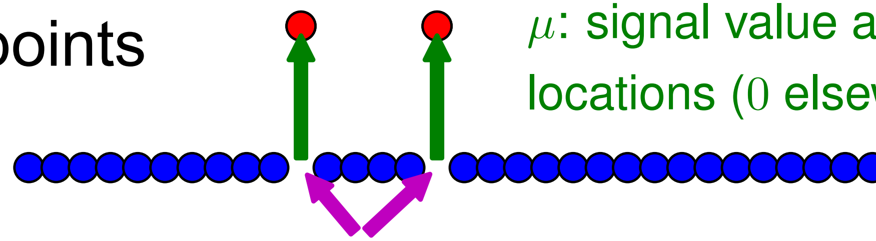
Objective: design x_t to recovery the support of β^*

Not Compressive Sensing Or Distilled Sampling

1d search, n total grid points

k-sparse signal

$$\boldsymbol{\beta}^* \in \mathbb{R}_+^n$$



aggregate measurement

$$\mathbf{x}_t \in \mathbb{R}_+^n, \ \|\mathbf{x}_t\|_2 = 1$$

Sensing model

$$y_t = \mathbf{x}_t^\top \boldsymbol{\beta}^* + \varepsilon_t, \ \varepsilon_t \sim \mathcal{N}(0,1)$$

Objective: design x_t to recovery the support of β^*

Binary Search

Goal: find 1-sparse signal with noiseless measurements

Binary search algorithm

Init: Set valid region to the entire environment

Repeat

Choose x_t to bisect the valid region

Observe y_t

Keep or eliminate the section corresponding to x_t

Until the valid region contains a single point

Total number of measurements:

$O(\log_2 n) = \tilde{O}(1)$, hiding logarithmic factors

Our Algorithm
Region Sensing Index (RSI)

For 1-sparse signal, assume uniform prior on

$$\boldsymbol{\beta} \in \{\mu\mathbf{e}_1, \mu\mathbf{e}_2, \ldots, \mu\mathbf{e}_n\}$$

Repeat

Maximize Information Gain (IG) $\arg\max_{\mathbf{x}_t} I_{t-1}\big(\boldsymbol{\beta}; y(\mathbf{x}_t)\big)$

Observe y_t

Update $p_t(\boldsymbol{\beta}) \propto p_{t-1}(\boldsymbol{\beta}) p(y_t \mid \mathbf{x}_t^\top \boldsymbol{\beta})$

Until $p_t(\boldsymbol{\beta})$ concentrates on a single point location

For k-sparse, repeat the above to find each signal or directly build distributions on all k-sparse signals

Noiseless Information Gain: Connection to Binary Search

Equivalent to marginal entropy,

$$I(\boldsymbol{\beta}; y(\mathbf{x})) = H(y(\mathbf{x})) - \underbrace{\mathbb{E}[H(y(\mathbf{x}) \mid \boldsymbol{\beta})]}_{\text{Const.}}$$

e.g., with noiseless measurements

$$y(\mathbf{x}) = \begin{cases} \frac{\mu}{\sqrt{\|\mathbf{x}\|_0}} & \text{if } \mathbf{x}^\top \boldsymbol{\beta}^* > 0; \\ 0 & \text{otherwise.} \end{cases}$$

$p_t(\mathbf{x}^\top \boldsymbol{\beta} > 0)$: hit chance, $\frac{\mu}{\sqrt{\|\mathbf{x}\|_0}}$: measurement strength.

Maximum IG is log(2), by binary search, $p_t(\mathbf{x}^\top \boldsymbol{\beta} > 0) = \frac{1}{2}$.

Information Gain with Noise

Equivalent to marginal entropy,

$$I(\boldsymbol{\beta}; y(\mathbf{x})) = H(y(\mathbf{x})) - \underbrace{\mathbb{E}[H(y(\mathbf{x}) \mid \boldsymbol{\beta})]}_{\text{Const.}}$$

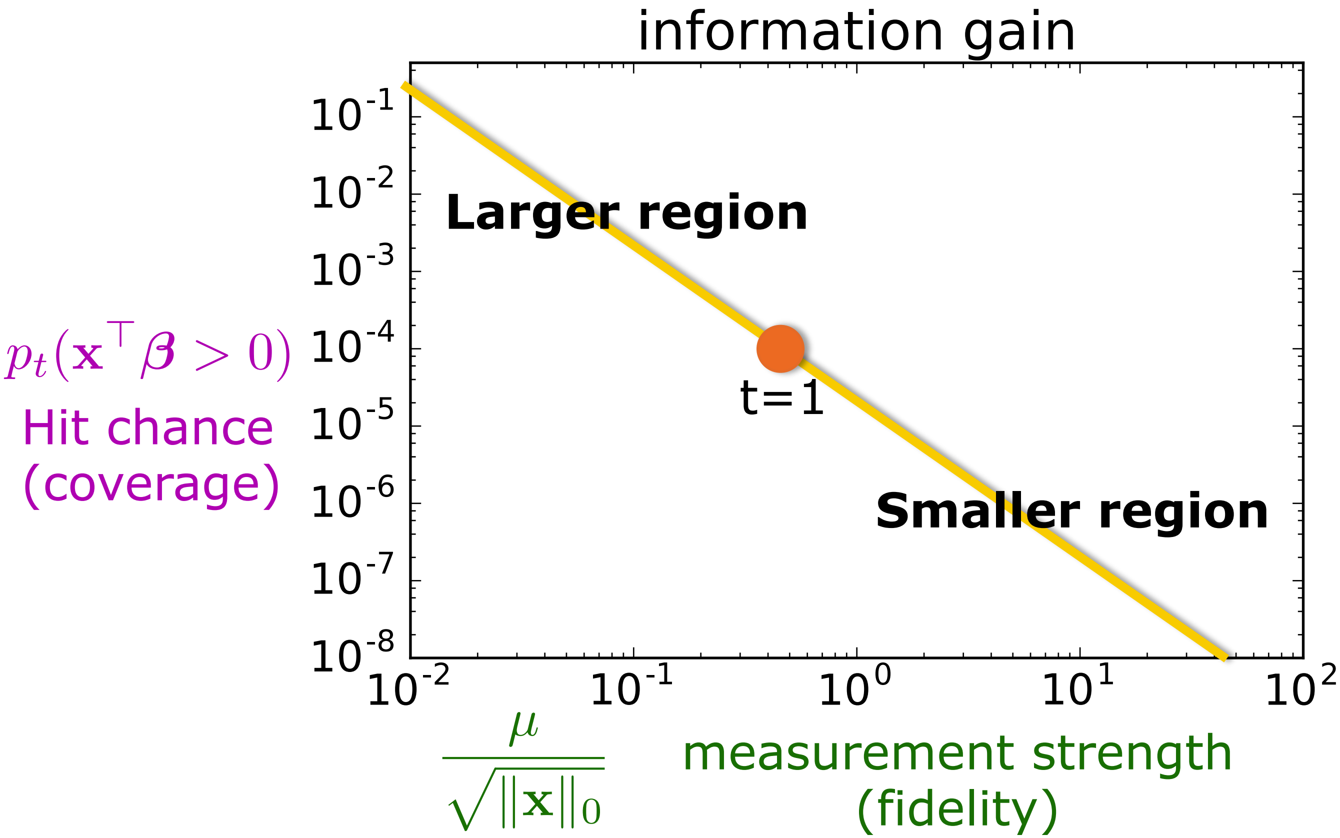
e.g., with noisy measurements

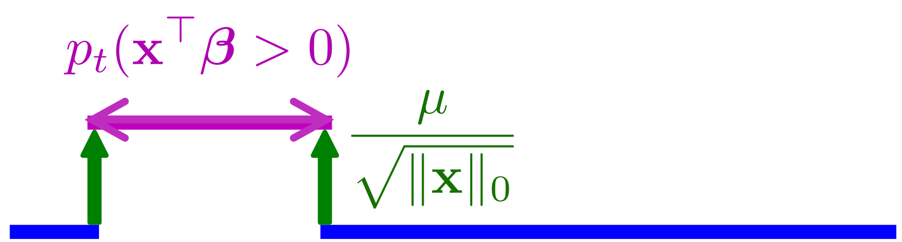
$$y(\mathbf{x}) \sim \begin{cases} \mathcal{N}\left(\frac{\mu}{\sqrt{\|\mathbf{x}\|_0}}, 1\right) & \text{if } \mathbf{x}^\top \boldsymbol{\beta}^* > 0; \\ \mathcal{N}(0, 1) & \text{otherwise.} \end{cases}$$

$p_t(\mathbf{x}^\top \boldsymbol{\beta} > 0)$: hit chance, $\frac{\mu}{\sqrt{\|\mathbf{x}\|_0}}$: measurement strength.

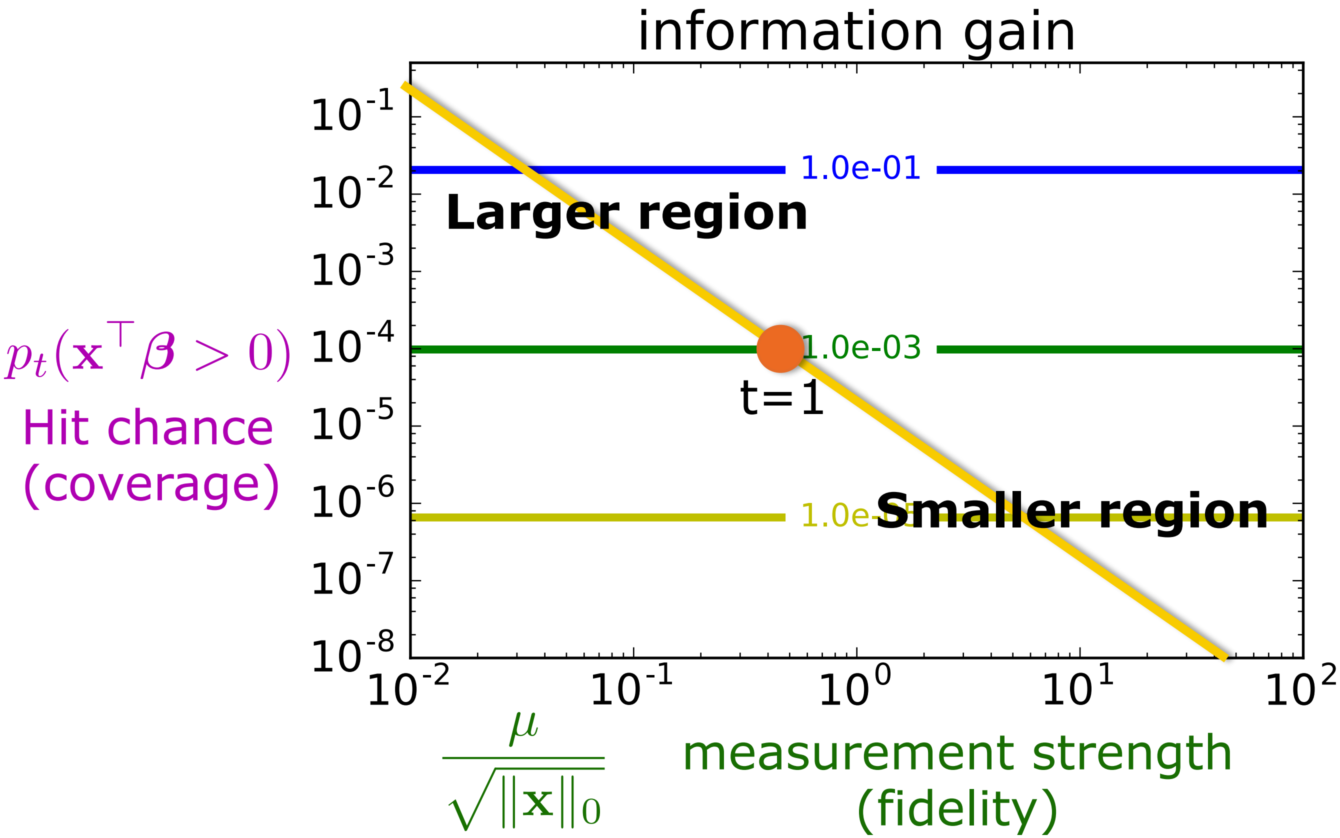
Maximum IG is less than log(2), but how much less?

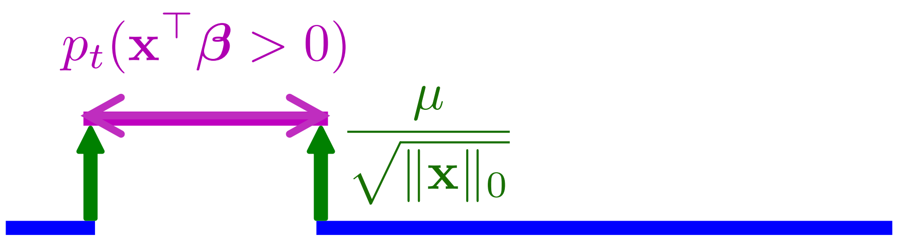
Coverage vs. Fidelity



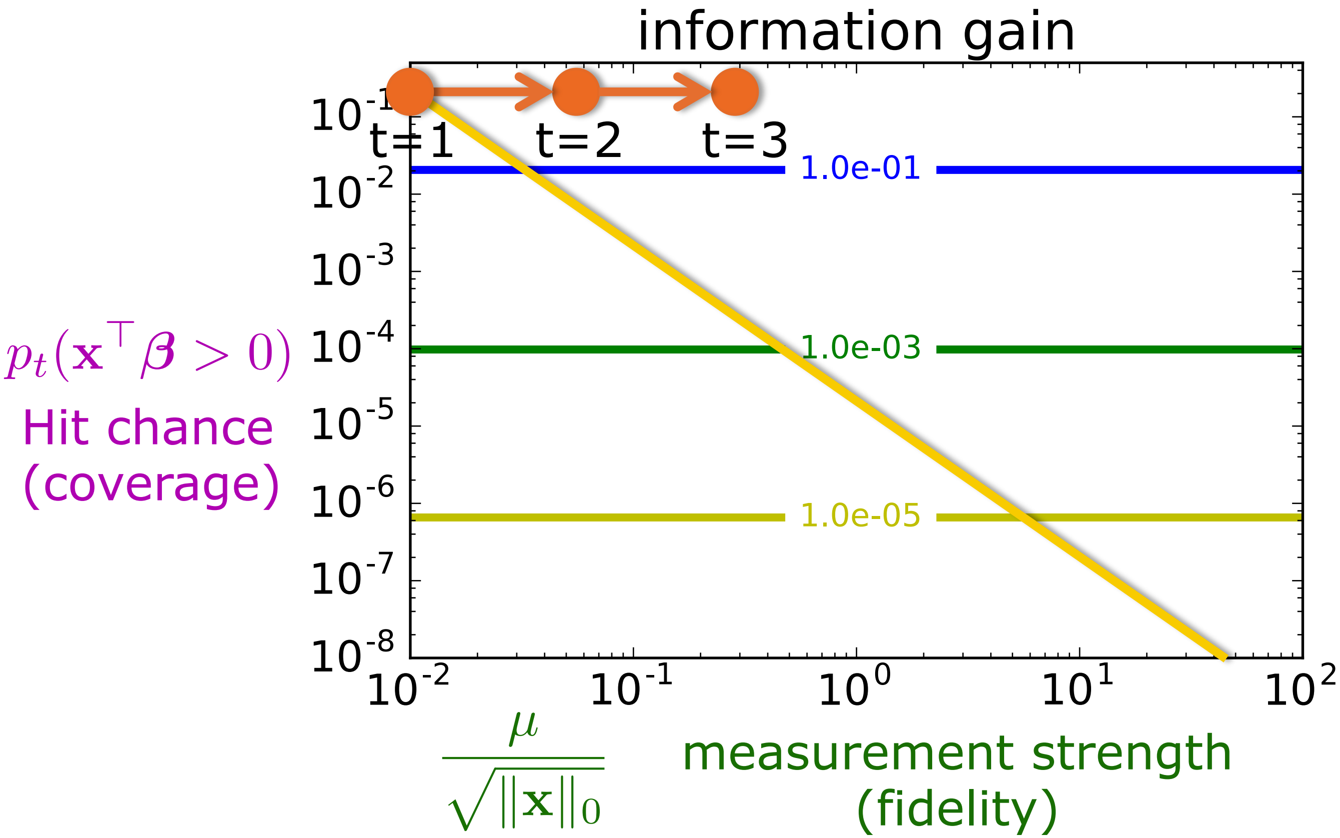


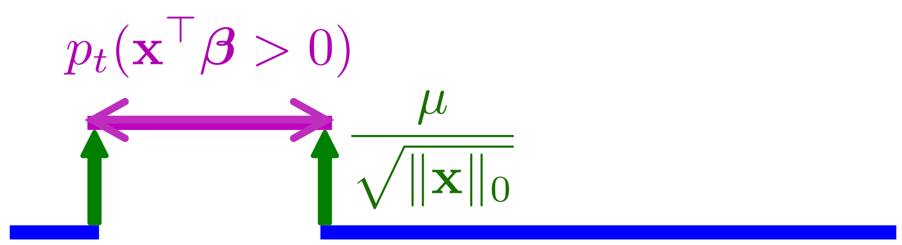
Noiseless IG Contour



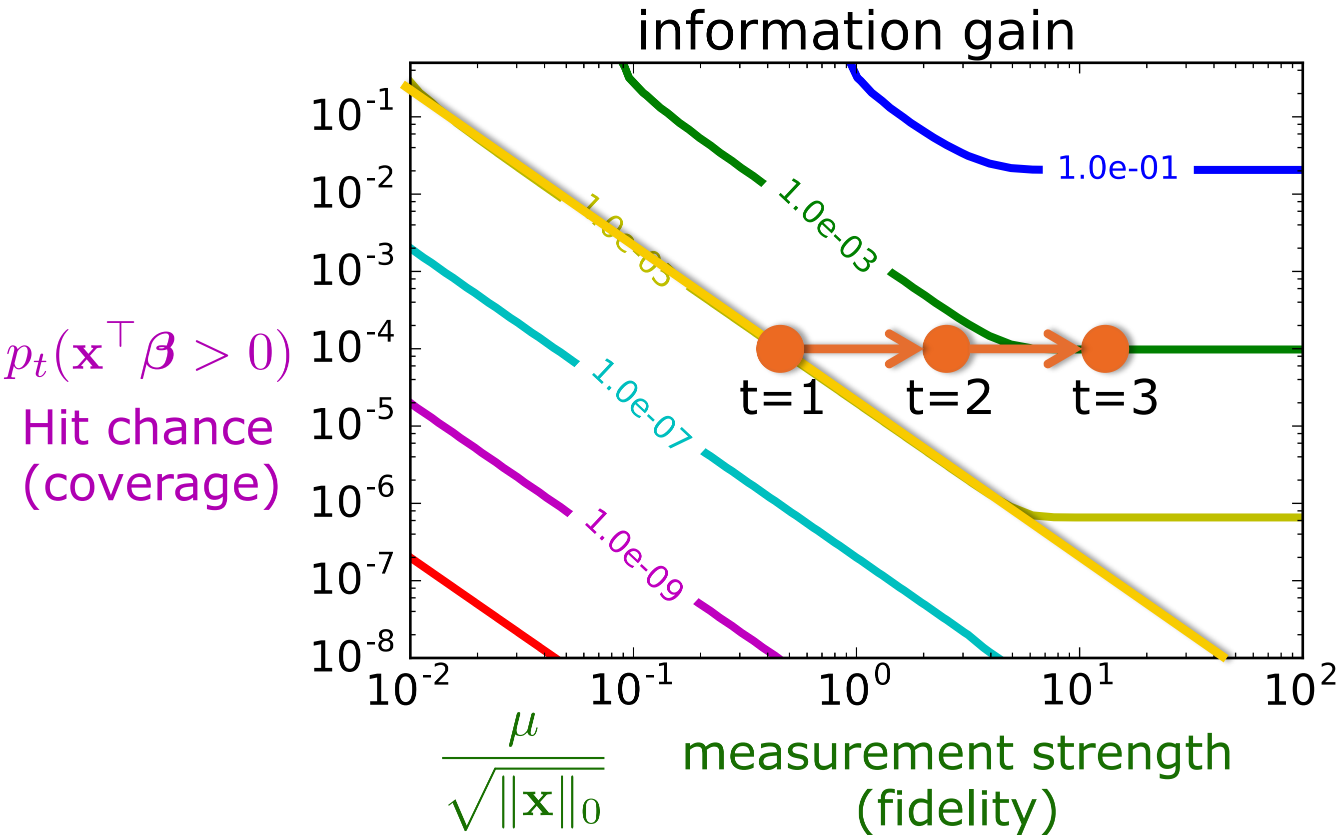


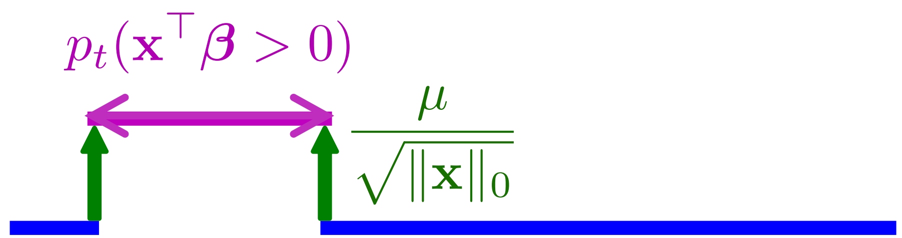
Noiseless IG Contour





IG Contour with Noise





Number of Measurements

Before finding the true signal, for every query

$$I(\boldsymbol{\beta}, y(\mathbf{x})) \geq \min\left\{\frac{\mu^2}{12n}, \frac{1}{8}\right\}$$

Because a uniform prior on β has no more than log(n) bits of uncertainty, the expected number of measurements is at most

$$\tilde{O}\left(\frac{n}{\mu^2} + k^2\right)$$

Larger signal-to-noise ratio μ => fewer measurements

Near-optimal rate

Simulation Result

Fix search space (d=1, n=1024) and 1-sparse signal

As we vary signal-to-noise ratio μ, the number of measurements change

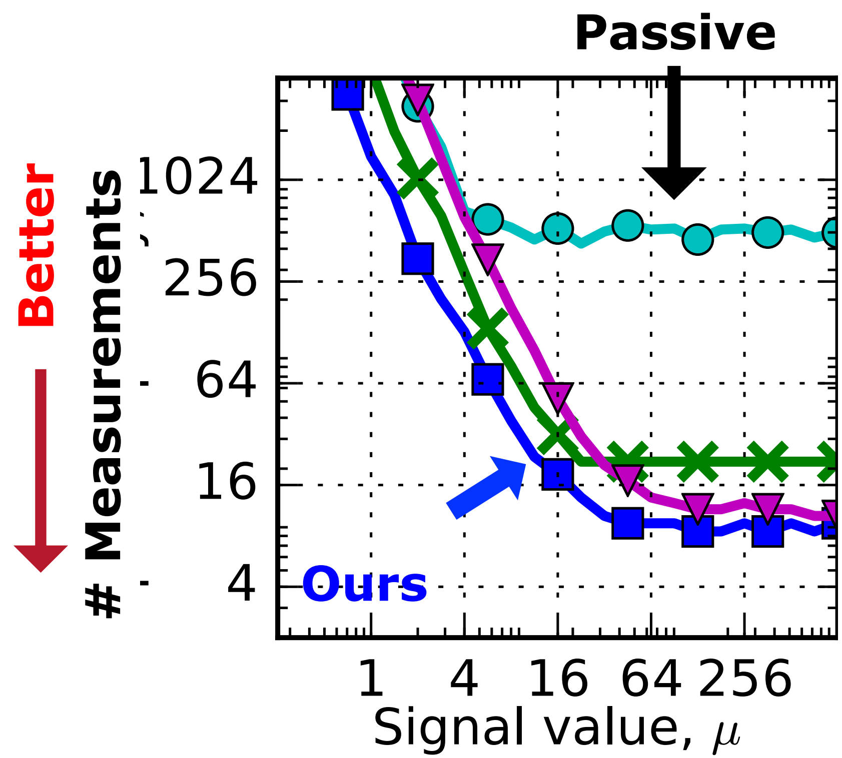
At same μ, our method uses the fewest number of measurements

RSI: Our algorithm

CASS*: Malloy&Nowak 2013

Point: point sensing

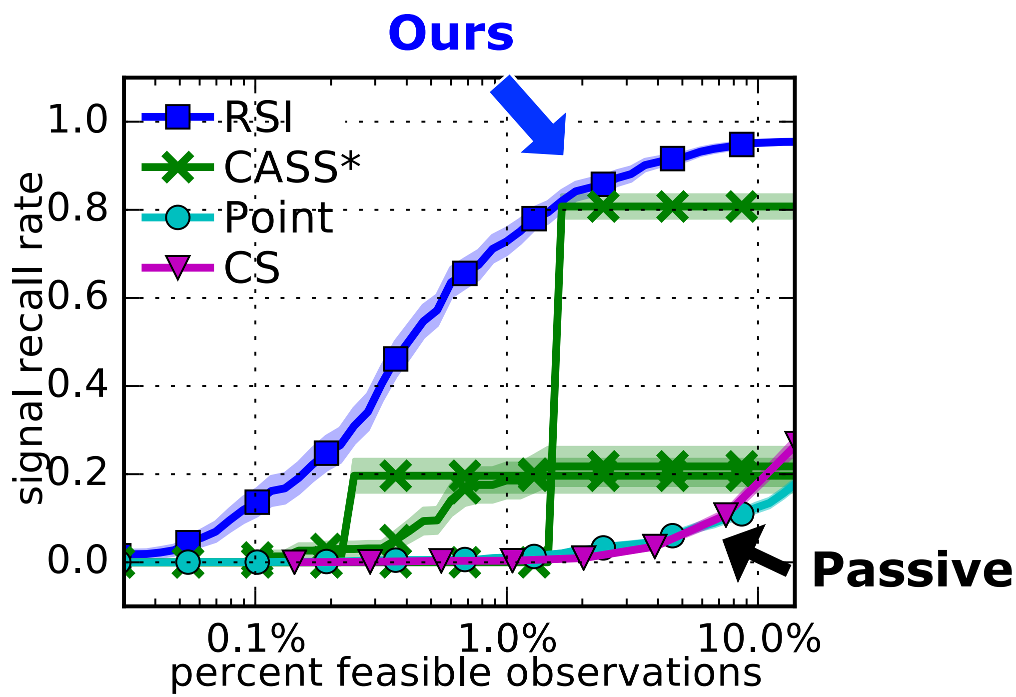
CS: Compressive sensing



Search for Blue Pixels on Satellite Images

Satellite images like the demo

Our method finds the most number of blue pixels w/ equal observations.



Summary: Active Aerial Search

Allow queries on a region of points

- Only average value is kept
- Coverage vs. fidelity trade-off
- Propose algorithm RSI by information criteria
- Near-optimal expected number of measurements
- Experiments with real satellite images

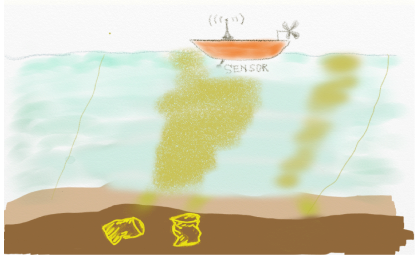
Outline / Contributions

Active search on graphs

- (NIPS 2013; UAI 2015)

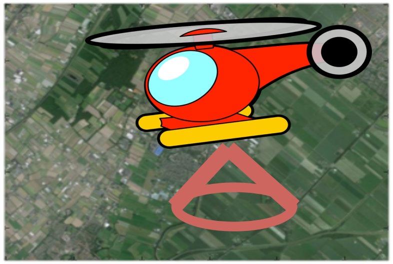
Active search with region rewards

- (AISTATS 2014;2015)



Active search with region queries

- (AAAI 2017)



Fast active search using conjugate sampling

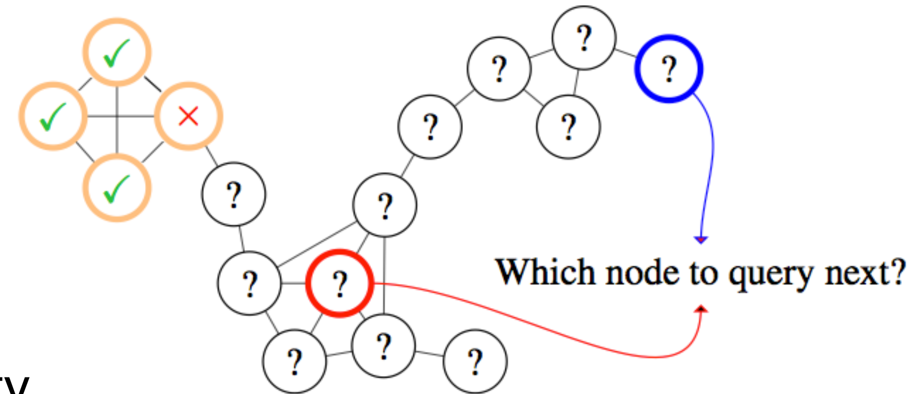
- (in preparation)

Scalability Issues

Bayesian methods are …

"Optimal" for Designs	Notoriously Slow	Memory Intense
Graphs with n nodes	$O(n^3)$ initial, then $O(n^2)$	$O(n^2)$
GP with n points	$O(n^2)$ per step	$O(n^2)$
Aerial search with k signals	$O(n^k)$ per step	$O(n^k)$

Thompson Sampling



Recall active search on graphs

Exploration: reduce model uncertainty

Plus Exploitation: check likely positives to collect rewards

Thompson sampling

Sample $\tilde{\mathbf{f}} \sim \mathcal{N}\left(\mathbb{E}(\mathbf{f} \mid \mathbf{y}_S), \mathrm{Cov}(\mathbf{f} \mid \mathbf{y}_S)\right)$

Pick $s_{t+1} = \arg\max_i \tilde{f}_i$

How to sample efficiently?

Exact Sampling from Multivariate Normal Distributions

The usual approach

In order to draw $\tilde{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$

Step 1. Decompose $\mathbf{C} = \mathbf{P}\mathbf{P}^\top$

Step 2. Draw iid $\xi_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$

Step 3. Transform $\tilde{\boldsymbol{\theta}} = \mathbf{P}\boldsymbol{\xi}$

Can we make it faster when C = A^{-1} and A is sparse?

Complexity:

Gradient descent < solving linear systems < matrix decomposition

Conjugate Sampling (in preparation)

Goal: approximately sample from $\tilde{\theta} \approx \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1})$

1. **Use *k* conjugate gradient steps to solve**

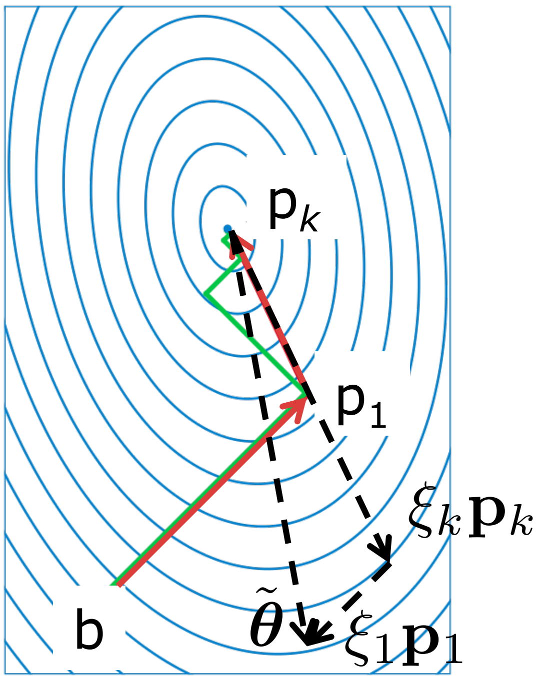
$$\mathbf{A}\mathbf{x} = \mathbf{b}, \text{ where } \mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

   Let the conjugate gradients be $p_1,\ldots,p_k$

2. **Keep a running sum**

$$\tilde{\boldsymbol{\eta}} = \sum_{i=1}^{k} \xi_i \mathbf{p}_i, \text{ where } \xi_i \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$$

3. **Rescale when k<n,** $\tilde{\boldsymbol{\theta}} = \sqrt{\frac{n}{k}}\tilde{\boldsymbol{\eta}}$



[Plot adapted from wikipedia by Oleg Alexandrov]

Exact Sampling When k=n

Conjugate vectors are A-orthogonal, we have

$$\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases}$$

Let P=($\mathbf{p}_1$, $\mathbf{p}_2$, …, $\mathbf{p}_n$),

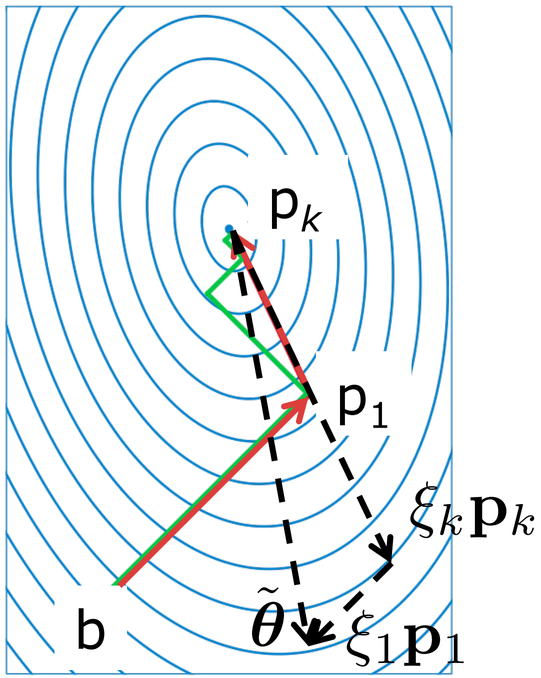
$$\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{I}$$

$$\mathbf{A} = \mathbf{P}^{-\top} \mathbf{P}^{-1}$$

$$\mathbf{A}^{-1} = \mathbf{P} \mathbf{P}^\top$$

Therefore,

$$\tilde{\boldsymbol{\theta}} = \mathbf{P} \boldsymbol{\xi} = \sum_{i=1}^{n} \xi_i \mathbf{p}_i$$



[Plot adapted from wikipedia by Oleg Alexandrov]

Intuition When k=1

Effectively, sample any

$$\mathbf{b} \sim \mathcal{N}(0, 1)$$

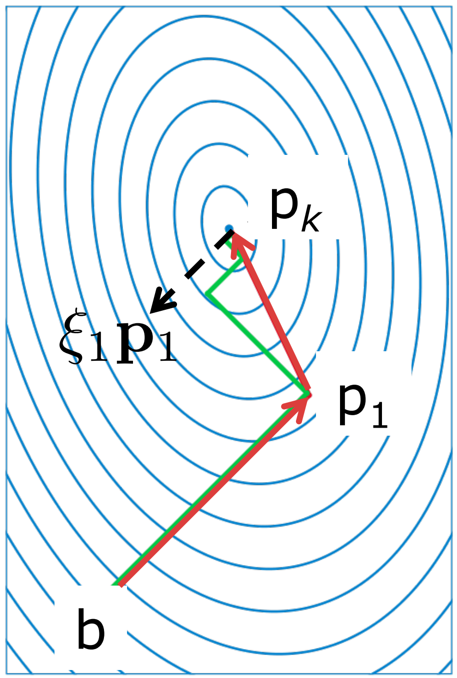
(A-)normalize to a unit direction vector

$$\mathbf{p}_1 = \frac{\mathbf{b}}{\sqrt{\mathbf{b}^\top \mathbf{A} \mathbf{b}}} = \frac{\mathbf{b}}{\|\mathbf{b}\|_\mathbf{A}}$$

Explore on the same direction with normalized scales

$$\tilde{\theta} = \sqrt{n}\xi\mathbf{p}_1, \text{ where } \xi \in \mathcal{N}(0, 1)$$

Exploration may be suboptimal, but sufficient in our simulations.



[Plot adapted from wikipedia by Oleg Alexandrov]

Simulation on Cumulative Regret

Show cumulative regret $\sum_{\tau=1}^{t}[f(x^*) - f(x_\tau)]$

Linear function $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$

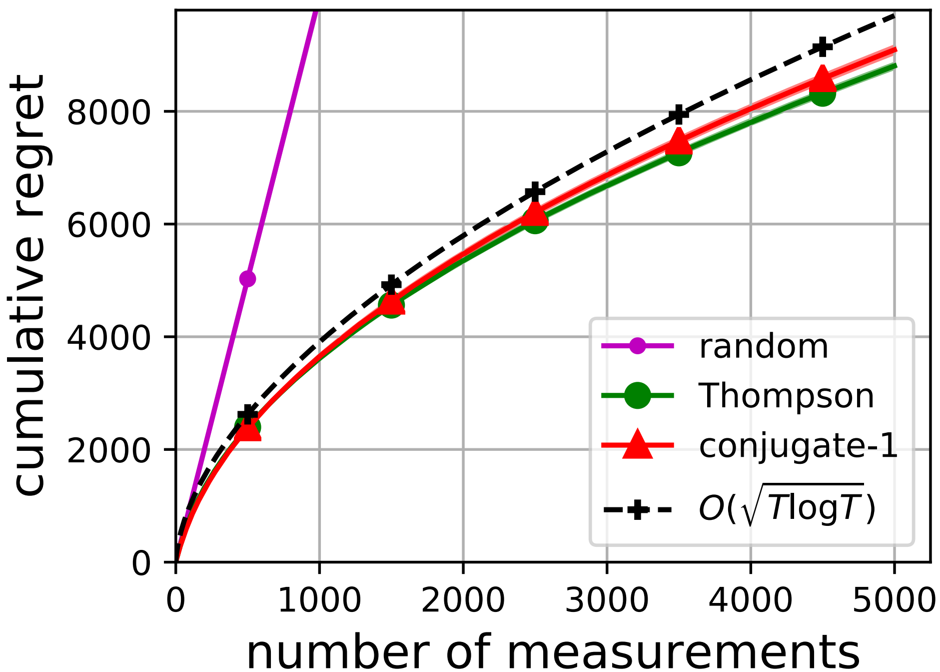
Choose any

$\mathbf{x} \in \mathbb{R}^{100}$ s.t. $\|\mathbf{x}\|_2 \leq 1$

n = 100

Unknown $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Observation noise = 1



Simulation on Cumulative Regret

Show cumulative regret $\sum_{\tau=1}^{t}[f(x^*) - f(x_\tau)]$

Smooth function $f(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{\text{SE}})$

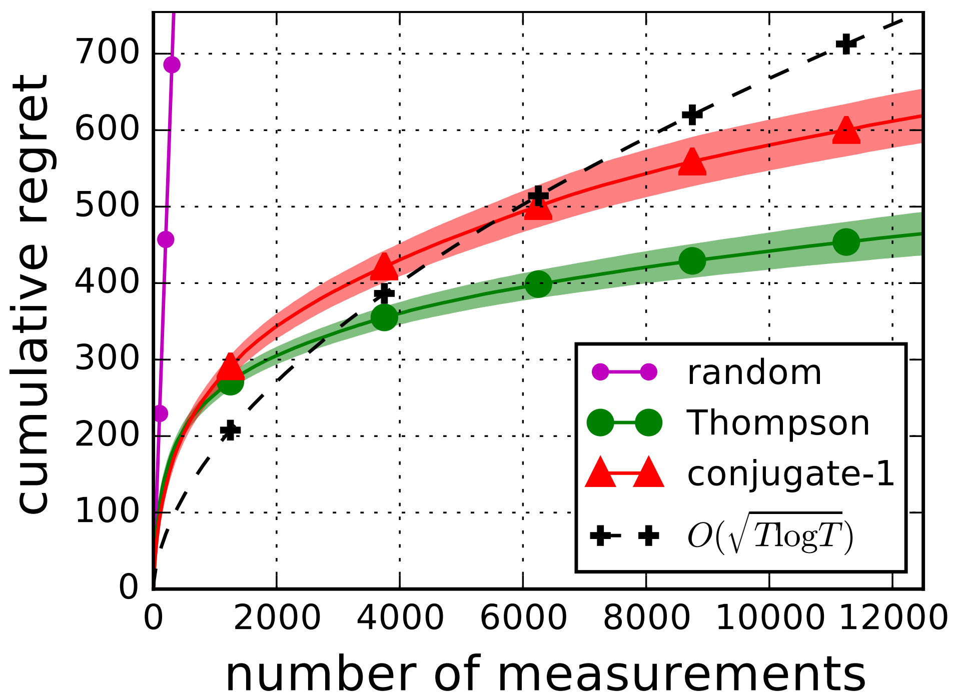
Choose any

$$\mathbf{x} \in \left\{0, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}, 1\right\}^3 \subset \mathbb{R}^3$$

n = 125

Kernel length-scale = 0.3

Observation noise = 1



Comparison

Assume information matrix, **A**, is ***n*-by-*n*** with ***m*** nonzero elements.

Assume $k \ll n$.

Method and condition	Time complexity (order)	Space complexity (order)
Thompson sampling (naïve)	n^3	n^2
Thompson sampling (online)	n^2	n^2
Rank-k matrix approximation	$k^2(n+m)$	$m+kn$
Rank-k conjugate sampling	$k(n+m)$	$m+n$

Disclaimer: still solves for the mean, the same order of complexity.

Conjugate Sampling Review

Bayesian methods are often slow to make decisions

- Thompson sampling draws only once to make greedy decisions
- Conjugate sampling aggressively approximates the posterior
- Faster designs on Graphs and Kronecker-GPs
- Similar regrets to exact Thompson sampling
- A lazy alternative for easy decision-making

Future work:

- On large graphs?
- Numerical stability?

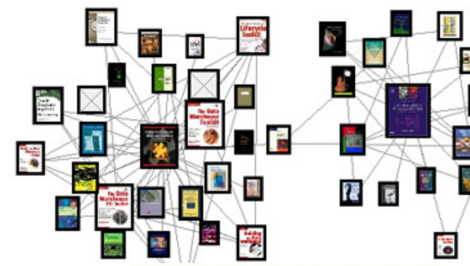
Auton Lab

Conclusion

Actively search for positives in an unknown environ by collecting and learning from feedback.

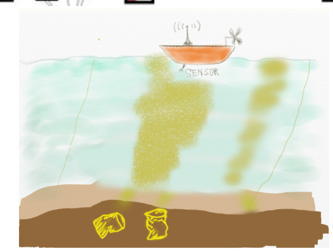
On graphs

- Σ-optimality as a better exploration heuristic
- Theoretical properties (global opt, cum. regret)



Region rewards

- Greedy maximization of expected rewards
- Point choices connect to Bayesian quadrature

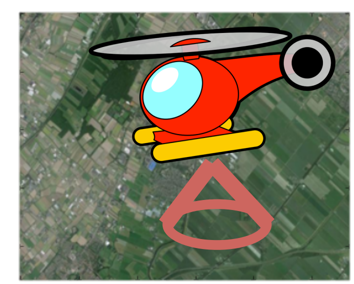


Region queries

- Extend binary search to noisy settings
- Bound expected number of measurements

Conjugate sampling

- Fast decision making for Graphs or GPs
- Flexible for more complex scenarios



Future Work

Active search on graphs

- Other models for node label distribution
- Exploration based on other spectral properties
- Use conjugate sampling for search on large graphs

Robotic applications

- Active search for areas of tumors, blood vessels, etc.
- Aerial search with multi-pixel camera
- Use reinforcement learning to imitate & improve search
- Path planning, ergodic exploration

Unified models for region queries and region rewards

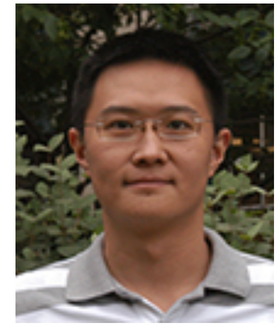
- Bipartite graph formulation
- Sampling based approach

Monte-Carlo tree search

- Games, combinatorial optimization, control with discrete states

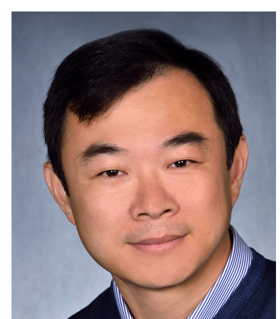
Acknowledgements







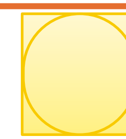
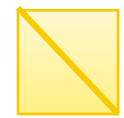
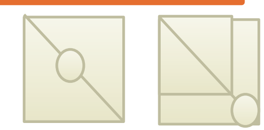
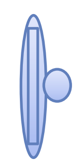




The Goal: Compare Sequential Active Learning Algos

Sequentially Select *s* for
$$\begin{cases} P(y_u|y_s) \propto \mathcal{N}(y_u; \hat{y}_u, L_u^{-1}) \\ L = \begin{pmatrix} L_u & L_{us} \\ L_{su} & L_s \end{pmatrix}, \quad \hat{y}_u = -L_u^{-1} L_{us} y_s \end{cases}$$

***s*: labeled, *u*: unlabeled. (*u*,*s*): complementary**

Possible strategies: (at step k with u^k unlabeled)

Σ-Optimality[1]	$\min_{v'} \left(\mathbf{1}^\top (L_{u^k \setminus \{v'\}})^{-1} \mathbf{1} \right)$	
V-Optimality[2]	$\min_{v'} \mathrm{tr} \left((L_{u^k \setminus \{v'\}})^{-1} \right)$	
Info Gain (IG)[3]	$\max_{v'} \left(L_{u^k}^{-1} \right)_{v',v'}$	
Mutual (MIG)[3]	$\max_{v'} \left(L_{u^k}^{-1} \right)_{v',v'} \Big/ \left((L_{\ell^k \cup \{v'\}})^{-1} \right)_{v',v'}$	
Uncertainty[4]	$\min_{v'} \lvert \hat{y}_{v'} \rvert$	
E Error (EER)[4]	$\max_{v'} \mathbb{E}_{y_{v'}} \left[\left(\sum_{u_i \in u} \lvert \hat{y}_{u_i} \rvert \Big\vert y_{v'} \right) \Big\vert y_{\ell^k} \right]$	

[1]Ma et. al. 2013. [2]Zhu et. al. 2003; Ji & Han, 2012. [3]Krause et. al. 2008. [4]Settles 2012.

Contributions

App	Challenge	Previous approach	Contribution	Papers
Information Retrieval	Similarity features	Linear models	Graphs	NIPS 2013; UAI 2015
Monitoring / Polling	Reward defined by a group of points	Point rewards	Group rewards	AISTATS 2014; 2015
Surveillance	Sparse signal	Point measurements	Aggregate measurements	AAAI 2017
Complex systems	Infeasible to find optimal design	Thompson sampling	Faster sampling	ICML 2017 Workshop

Alternative Intuition

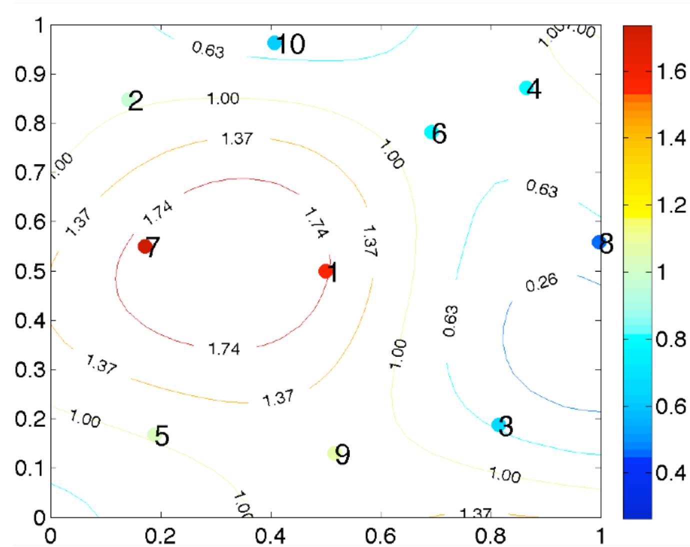
Assuming regions are independent

Select points in a region

Variance reduction of the integral

Bayesian quadrature [Minka 2000]

Σ-optimality

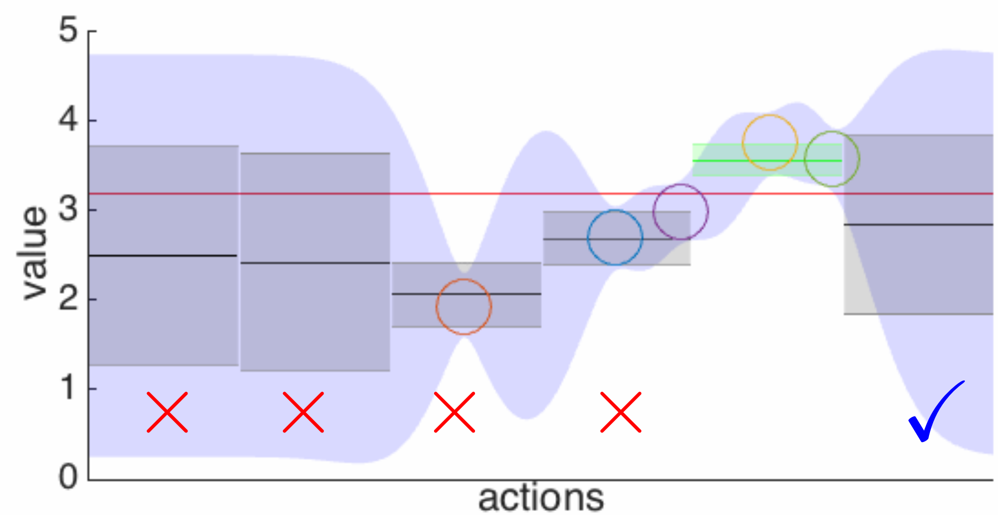


Select a region

High posterior mean

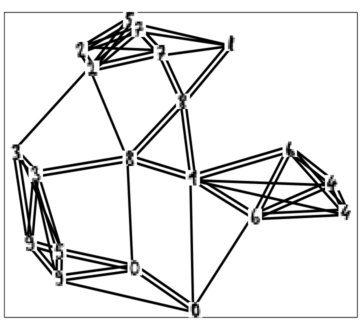
and

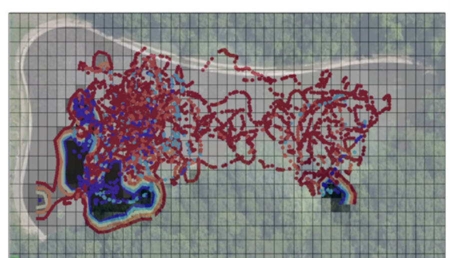
High variance reduction

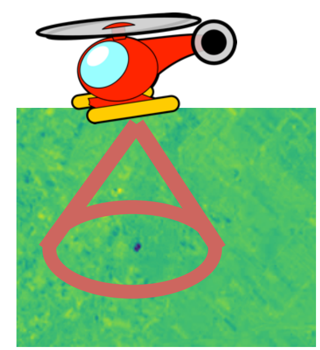


Summary

App	Challenge	Previous approach	Contribution	
Information Retrieval	Similarity features	Linear models	Graphs	NIPS 2013; UAI 2015
Monitoring / Polling	Reward defined by a group of points	Point rewards	Group rewards	AISTATS 2014; 2015
Surveillance	Sparse signal	Point measurements	Aggregate measurements	AAAI 2017
Complex systems	Infeasible to find optimal design	Thompson sampling	Faster sampling	ICML 2017 Workshop







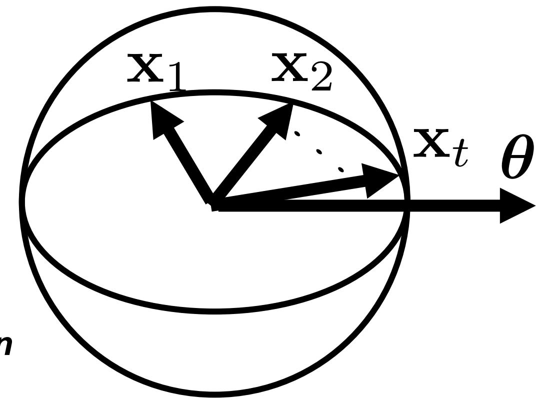
Linear Bandits in High-Dimensions

Problem: max $f(\bar{\mathbf{x}}) = \langle \bar{\mathbf{x}}, \theta \rangle$ **with unknown** $\boldsymbol{\theta}$

choose $\mathbf{x}_1, \ldots, \mathbf{x}_t$ **sequentially, s.t.** $\|\mathbf{x}_\tau\| \leq 1, \forall \tau$

observe $y_\tau = f(\mathbf{x}_\tau) + \varepsilon_\tau$

evaluate on $\bar{\mathbf{x}} = \frac{1}{t} \sum_{\tau=1}^{t} \mathbf{x}_\tau$



Intuitions:

To infer $\boldsymbol{\theta}$**,** $\mathbf{x}_{1:t}$ **must cover most directions in** R^n

To max f**,** $\mathbf{x}_t$ **must be close to empirical solution**

Challenges:

n **is large (**$n >> t$**)**

use prior information on $\boldsymbol{\theta}$

Problem Specification

Assume prior

$$p_0(\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{A}}_0^{-1})$$

Iterate

collect and infer posterior

$$p_t(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}^{-1})$$

where

$$\mathbf{A} = \bar{\mathbf{A}}_0 + \sum_{\tau<t} \mathbf{x}_\tau \mathbf{x}_\tau^\top$$

draw

pick $\tilde{\boldsymbol{\theta}} \sim p_t(\boldsymbol{\theta})$

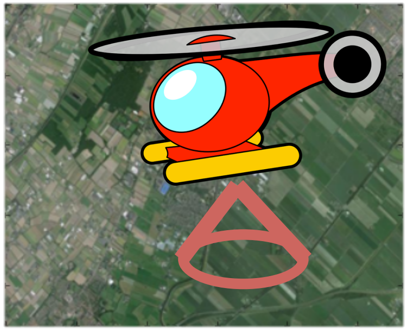
$$\mathbf{x}_t = \arg\max \langle \mathbf{x}, \tilde{\boldsymbol{\theta}} \rangle$$

- **Fast and approximate sample from?**

(ignore the mean) $p_t(\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1})$

Applications

Application	Active Search Allows
Environmental monitoring	Finding all polluted areas
Product recommendation	New users w/ little purchase history
Information retrieval	Relevant but underspecified results
Search and rescue	Localize all distress signals



Idea 1: Active Search on Graphs

Graphs can represent complex information

- High-dim sparse features, links, hierarchical structures.

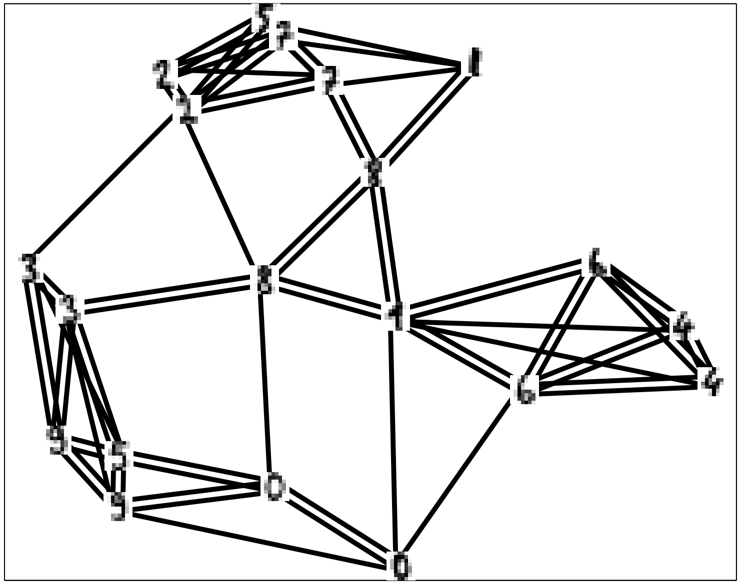
Important to predicting labels

Example:

A k-nearest-neighbor graph
of hand-written digits

Based on Euclidean distance
on concatenated pixel values

Visually similar digits form clusters



Idea 1: Active Search on Graphs

Graphs can represent complex information

- High-dim sparse features, links, hierarchical structures.

Important to predicting labels

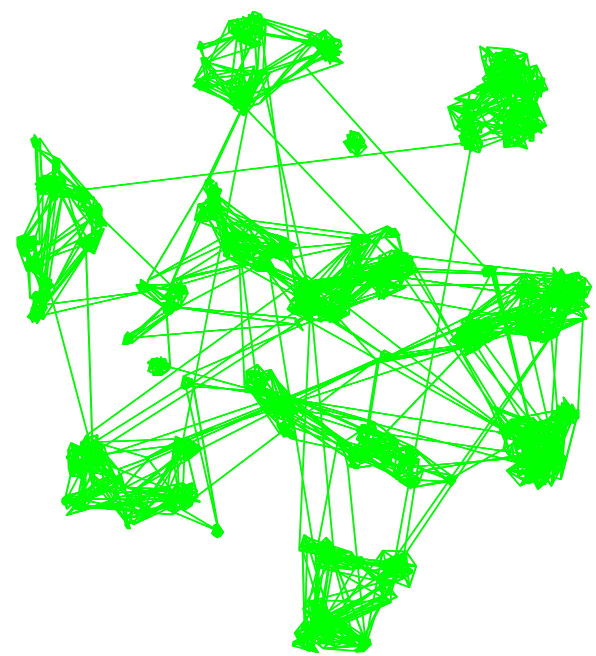
Example:

A k-nearest-neighbor graph
of hand-written digits

Based on Euclidean distance
on concatenated pixel values

Visually similar digits form clusters

Same graph with more data



Idea 1: Active Search on Graphs

Graphs can be important to label predictions

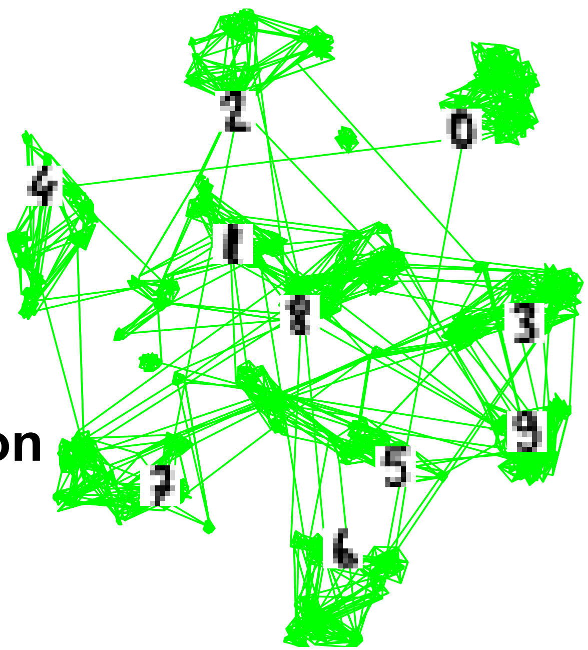
Example:

A k-nearest-neighbor graph
of hand-written digits

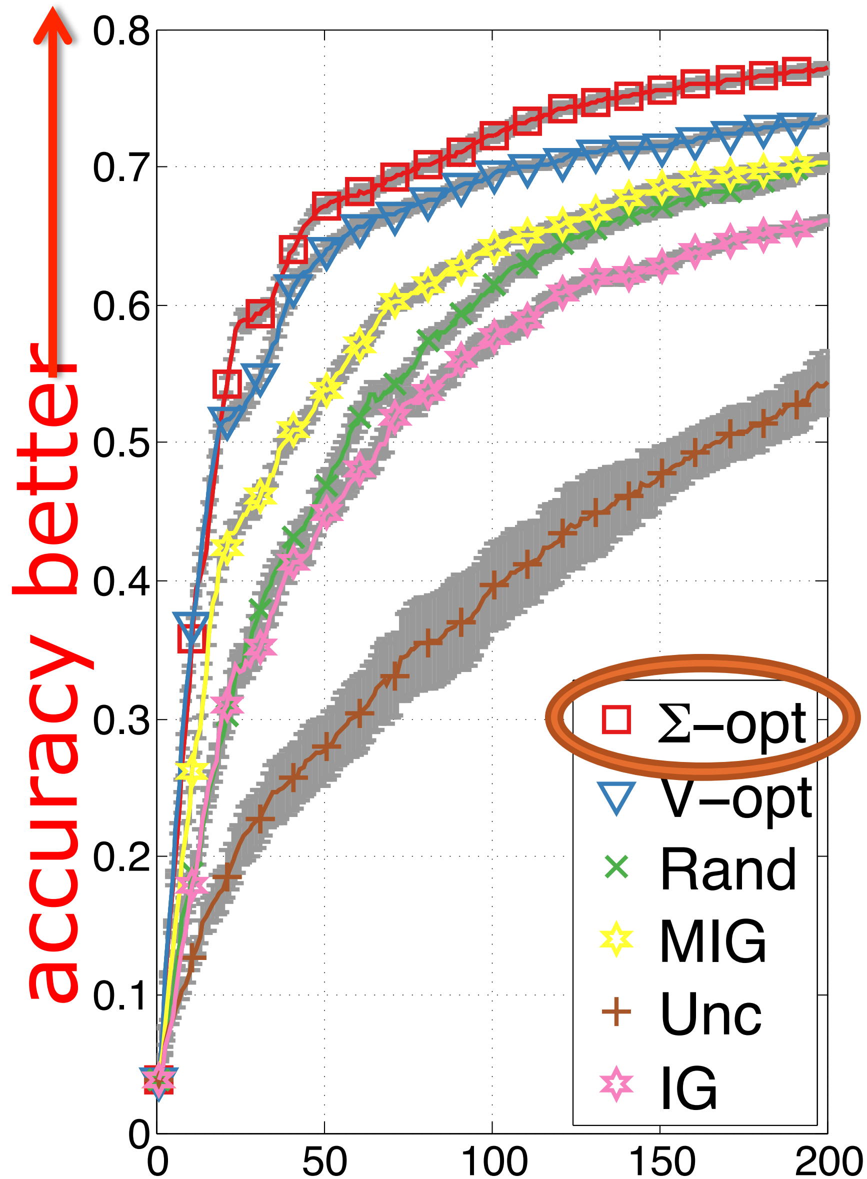
Based on Euclidean distance
on concatenated pixel values

Visually similar digits form clusters

A few queries often sufficient for prediction



Isolet 1+2+3+4



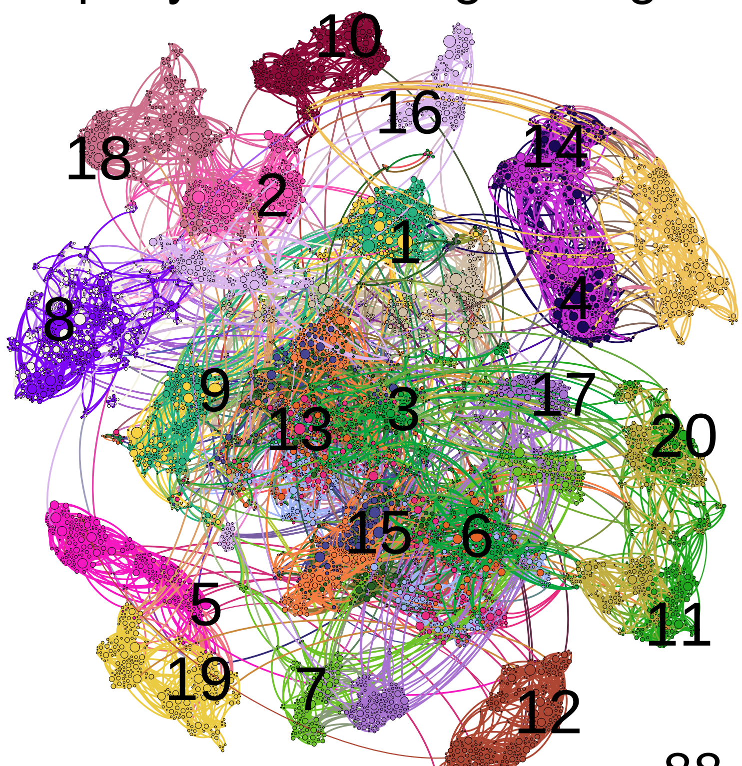
6238 Spoken letter recordings

617 dimensional frequency feature

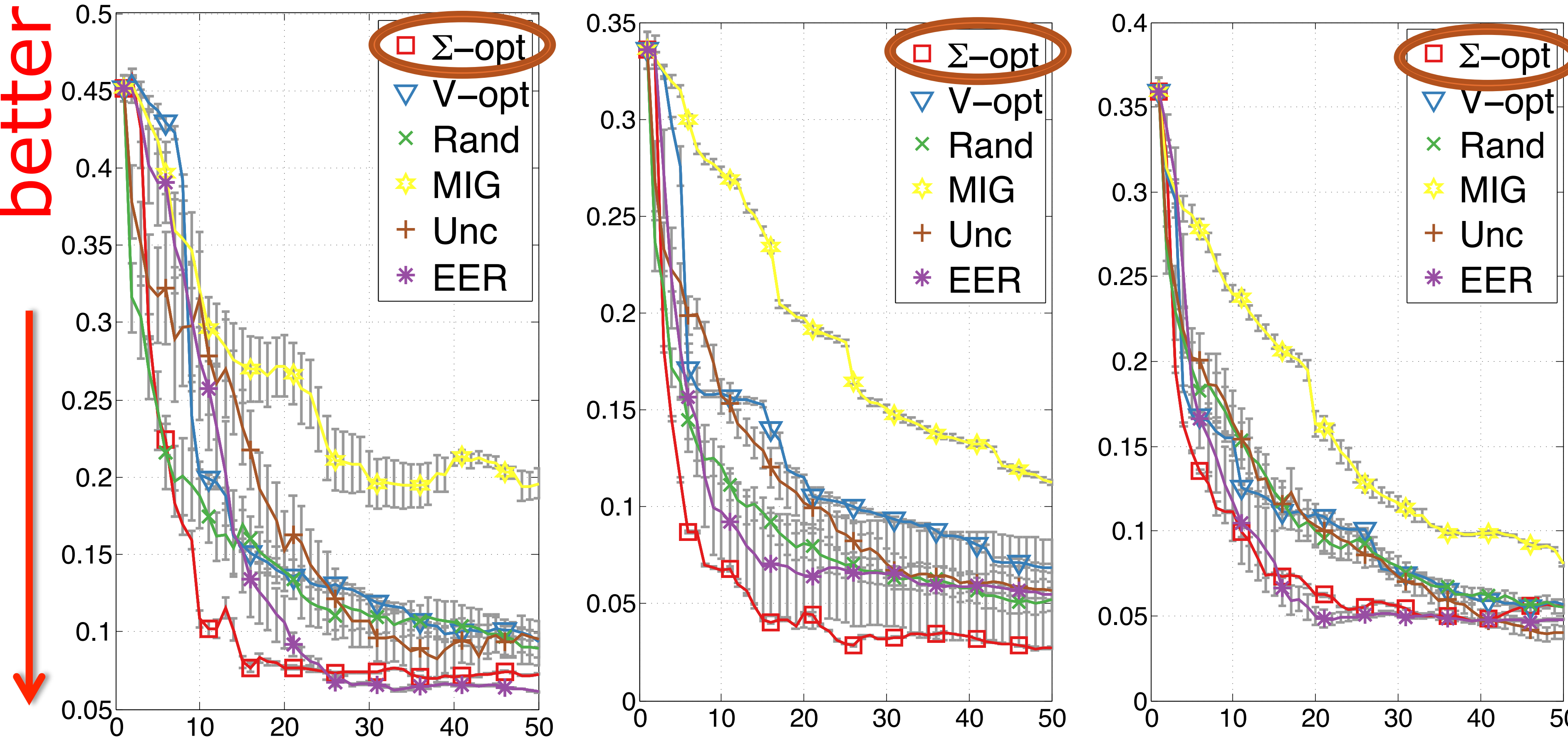
5-nearest neighbor graph from raw input

Random subsample 70% instances

First query fixed at largest degree



Active Surveying

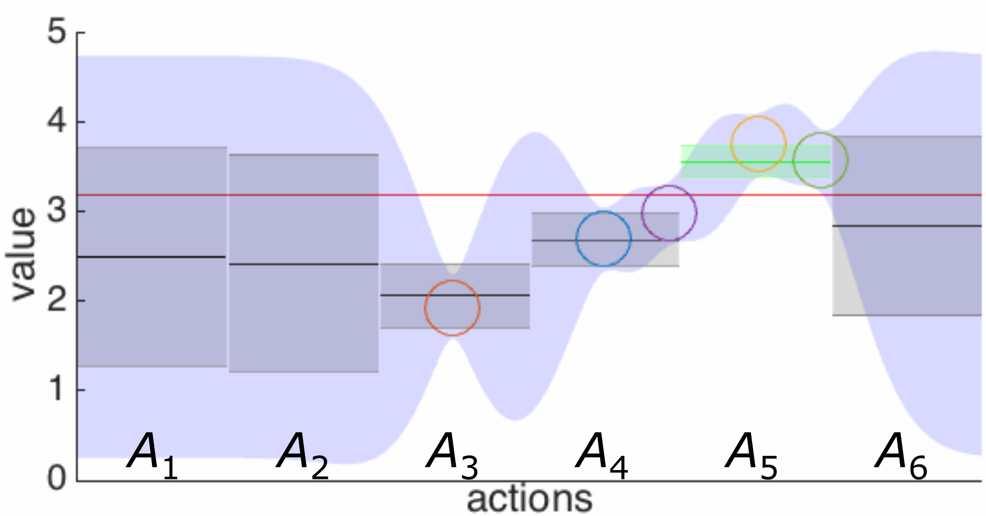


DBLP Coauthorship | Cora Citation | Citeseer Citation

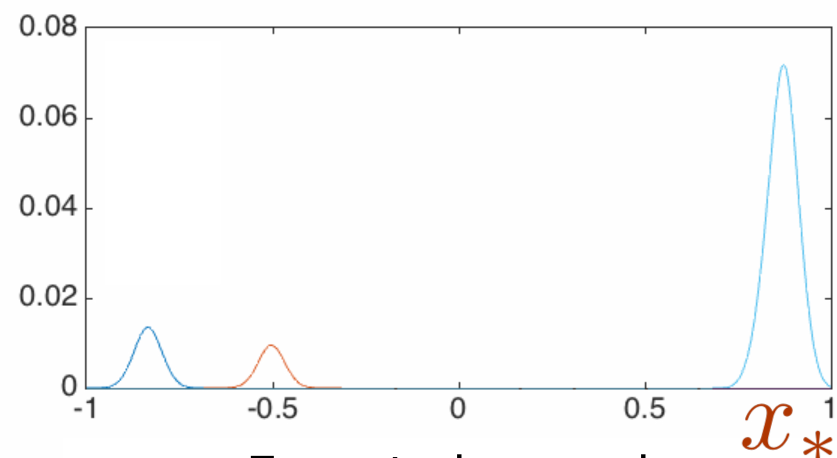
Algorithm

Maximize 1-step look-ahead expected reward

$$\max_{x_{t+1}} \int p_t(y_{t+1}|x_{t+1}) \cdot \sum_{g \in \mathcal{G}_t} \mathbf{1}(\mathrm{reward}_g \mid x_{1:t+1}, y_{1:t+1}) \, \mathrm{d}y_{t+1}$$



Posterior



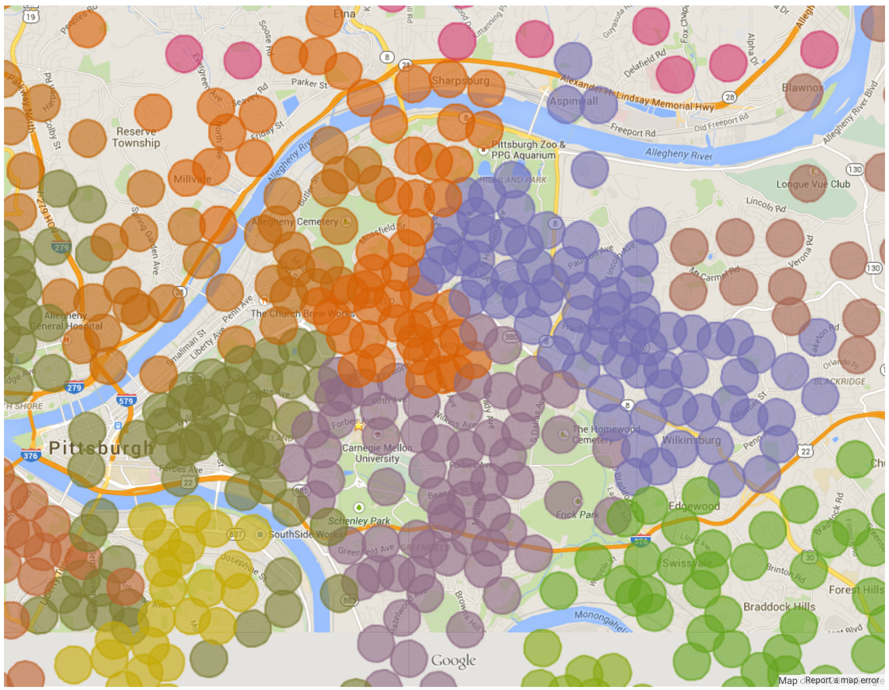
Expected reward

Circles: collected; blue: GP posterior; gray/green: post. of region integrals.

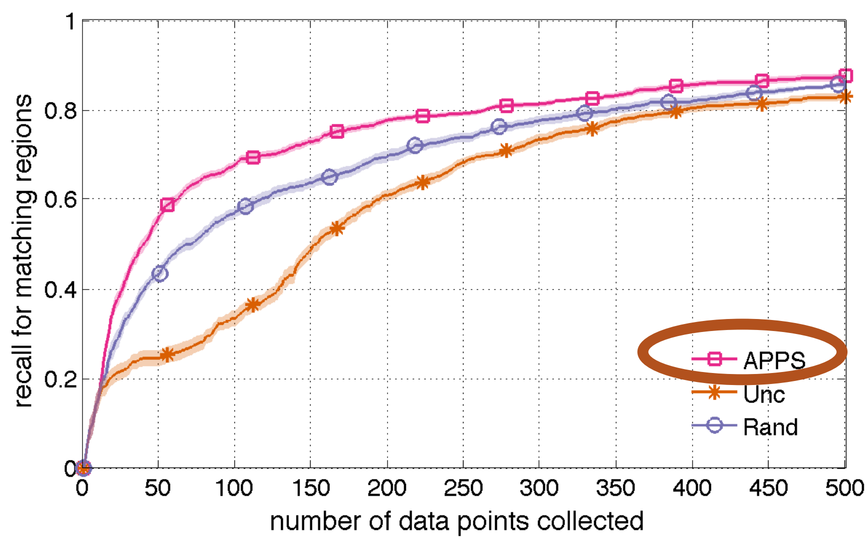
PA Election (Races vs. Precincts)

Search for positive electoral races with precinct queries.

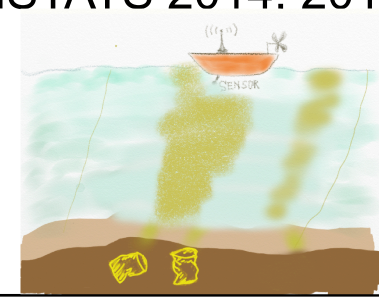
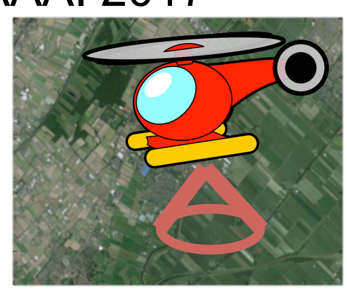
Regions



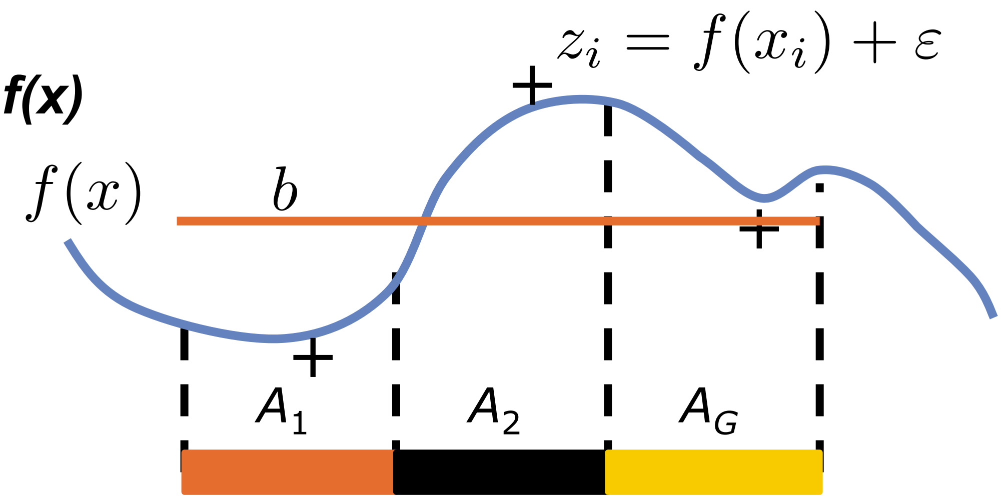
Dots: precinct centers, same color: races;
Build kernel on precincts by demographic info.



Outline / Contributions

Active search	Point rewards	Region rewards
Point queries	1. Active search on graphs NIPS 2013; UAI 2015 	2. Active area search AISTATS 2014: 2015 
Region queries	3. Active aerial search AAAI 2017 	• A unified model (future work) 4. Conjugate Sampling (in preparation)

Simple Pattern: Region Average



Assume a smooth function *f(x)*

Point observations

- Choose point x_i
- Observe value $z_i = f(x_i) + \varepsilon$

Region pattern

- Pre-define regions A_g for $g = 1, \ldots, G$
- Pattern: region average > a given value *b*

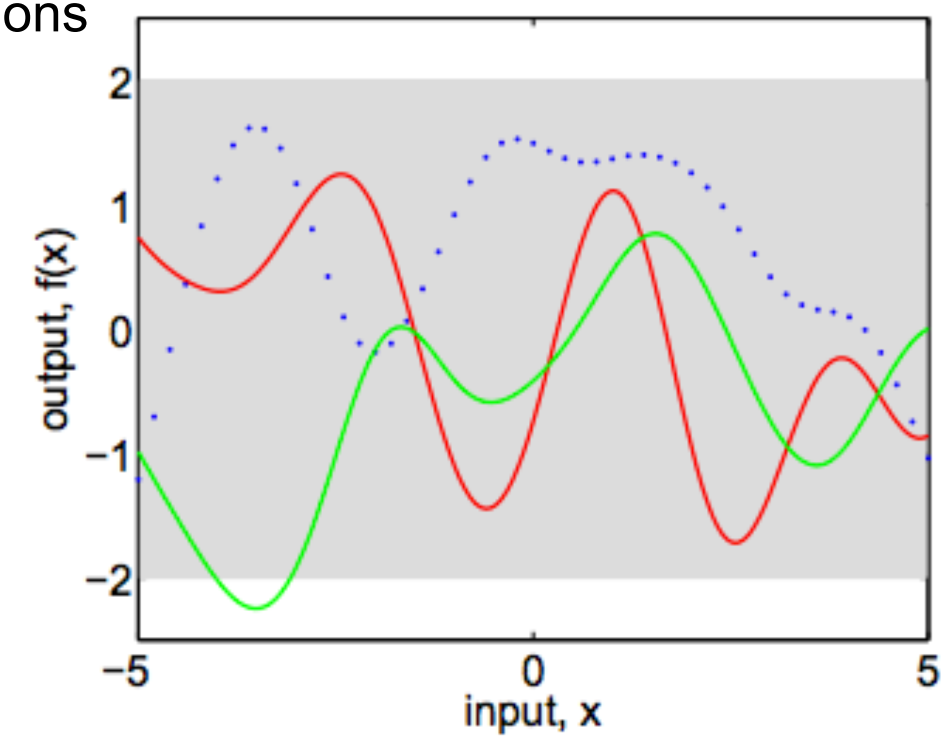
$$h_A(f) = 1_{\{\frac{1}{|A|}\int_A f \, \mathrm{d}x > b\}}$$

Point Observations Are Smooth

Assume *f(x)* a smooth function for *x* in R^d

⇔ Assume *f(x)* is drawn from a Gaussian Process (GP)

- A prior distribution over functions
- High prob. to draw smooth functions



Point Observations Are Smooth

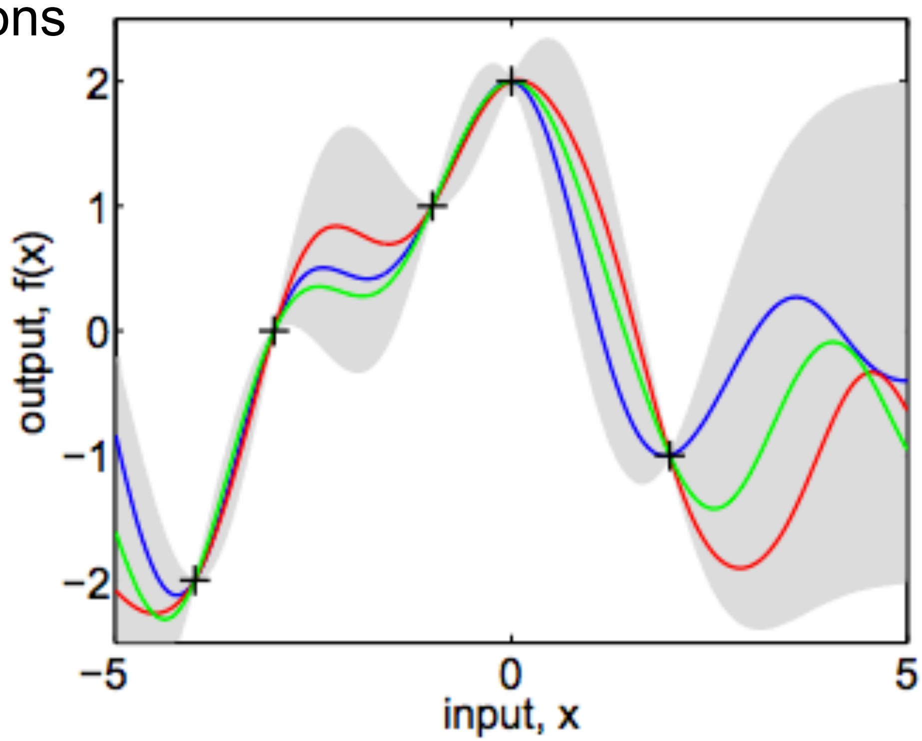
Assume *f(x)* a smooth function for *x* in R*^d*

⇔ Assume *f(x)* is drawn from a Gaussian Process (GP)

- A prior distribution over functions
- High prob. to draw smooth functions

With observed values ("+")

- Becomes a posterior dist.
- Consistent w/ observations

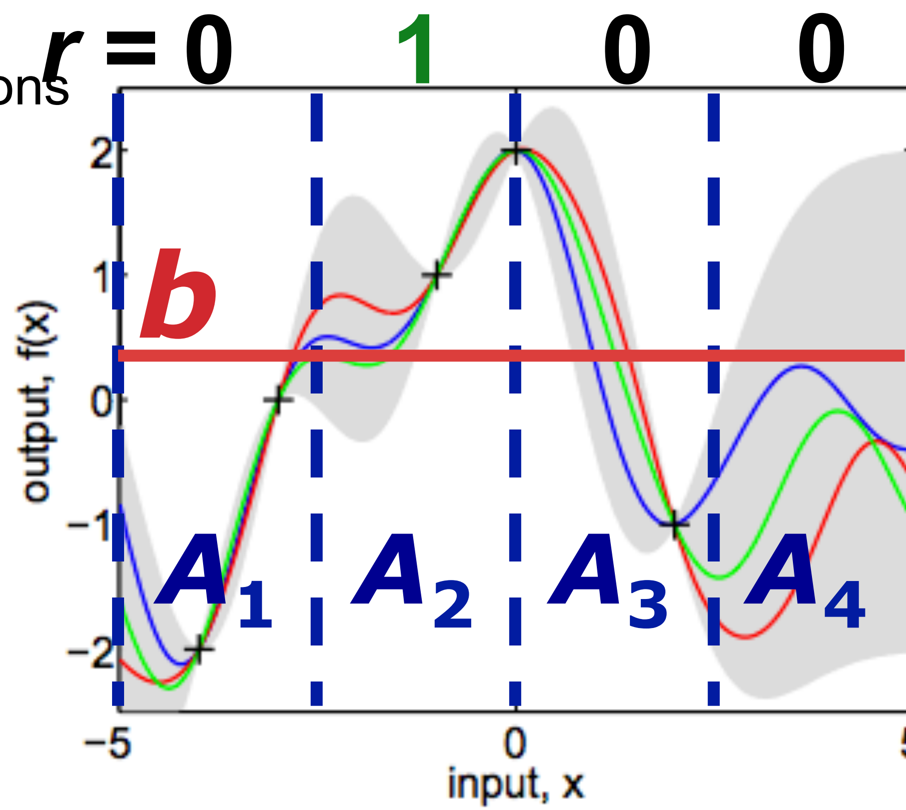


Point Observations Are Smooth

Assume *f(x)* a smooth function for *x* in R^d

⬄ **Assume *f(x)* is drawn from a Gaussian Process (GP)**

- A prior distribution over functions
- High prob. to draw smooth functions



With observed values ("+")

- Becomes a posterior dist.
- Consistent w/ observations

Assign rewards if region pattern has high probability

$$r_A(X, \mathbf{z}) = 1_{\left\{\mathbb{E}\left(h_A(f)|X,\mathbf{z}\right)>\theta\right\}}$$

Algorithm: Greedy Maximization of Expected Reward

Reward: $$r(X,\mathbf{z}) = \sum_{g=1}^{G} 1_{\left\{\mathbb{E}\left(h_{A_g}(f)|X,\mathbf{z}\right)>\theta\right\}}$$

At step t+1, choose location x_{t+1} to maximize

$$u(x_{t+1}) = \mathbb{E}\left[r(X_{1:t+1},\mathbf{z}_{1:t+1}) \;\middle|\; z_{t+1} \sim \mathcal{GP}(z(x_{t+1}) \mid X_{1:t},\mathbf{z}_{1:t})\right]$$

How? Use Bayesian look-ahead decisions

1. Sample possible outcomes from GP posterior

$$\tilde{z}_{t+1}^{(1)},\ldots,\tilde{z}_{t+1}^{(s)} \sim \mathcal{GP}(z(x_{t+1}) \mid X_{1:t},\mathbf{z}_{1:t})$$

2. For each $\tilde{z}_{t+1}$, estimate (by additional sampling if necessary)

$$\tilde{r} = r(X_{1:t}\cup\{x_{t+1}\},\mathbf{z}_{1:t}\cup\{\tilde{z}_{t+1}\})$$

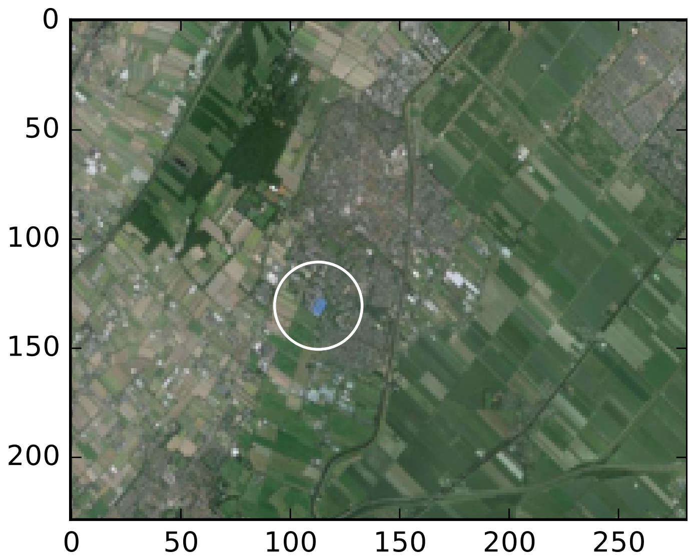
3. Estimate $$\hat{u}(x_{t+1}) = \frac{1}{s}\sum_{j=1}^{s}\tilde{r}^{(j)}$$

Demo Active Search

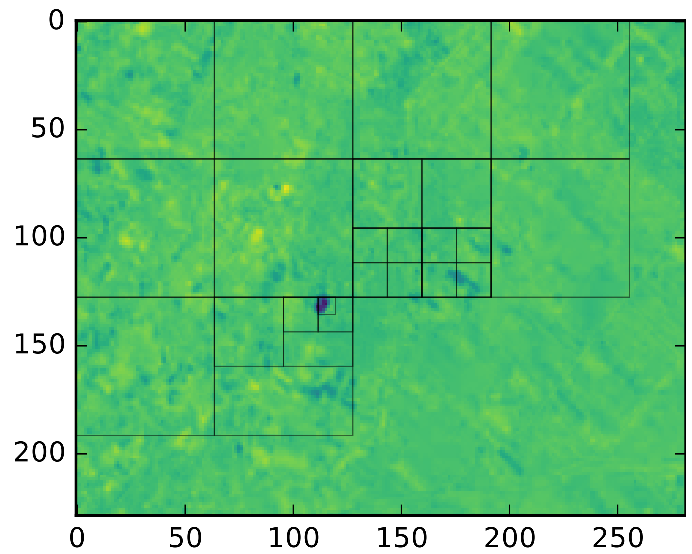
Find blue colors on a real satellite image

Simulate search and rescue in open areas

Used a blue filter on the RGB values, yielding scalar outcomes



(a) True point values



(b) search sequence