Adaptive Sensing for Recovering Structured Sparse Sets

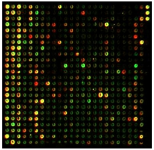
Ervin Tánczos, Rui Castro
Eindhoven University of Technology

Structures Seminar 16.10.2015

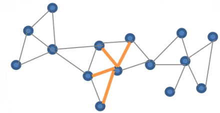
Support Recovery/Detection

Motivation

We are interested in recovering the support (or detecting the presence) of an unknown signal.



Gene expression



Network anomalies

Astronomical observations

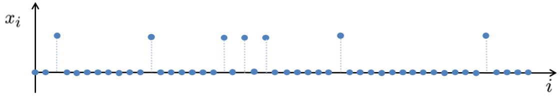
Support Recovery/Detection

Classical Framework

Let $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ denote the unknown signal where

$$x_i = \begin{cases} \mu & , \text{if } i \in S \\ 0 & , \text{if } i \notin S \end{cases},$$

with $\mu > 0$ fixed and $S \in \mathcal{C}$ where $\mathcal{C}$ is a class of sets.

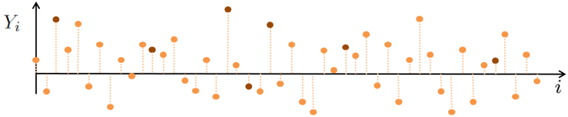


Support Recovery/Detection

Classical Framework

We observe

$$Y_i = x_i + W_i, \;\; W_i \overset{iid}{\sim} N(0,1), \;\; i = 1, \ldots, n \; .$$



Our goal is to recover S or to detect its presence (decide between $H_0 : \;\; S = \emptyset$ and $H_1 : \;\; \emptyset \neq S \in \mathcal{C}$).

How does μ need to scale so that the above tasks are possible?

Support Recovery/Detection

Classical Framework

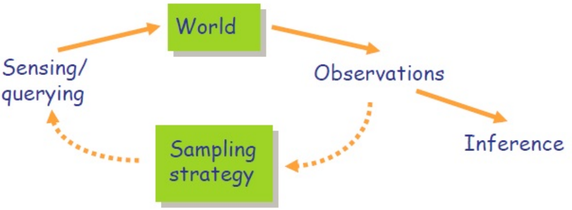
Depends on the class $\mathcal{C}$. From now on, assume $|S| = s \ll n \; \forall S \in \mathcal{C}$ (sparse signals). We want $\max_{S\in\mathcal{C}} \mathbb{P}(Error)$ to be small.

Non-Adaptive	Detection	Recovery
s-sets	$\sqrt{\log n}$	$\sqrt{\log n}$
unions of k disjoint s-intervals	$\sqrt{\frac{1}{s}\log n}$	$\sqrt{\frac{1}{s}\log n}$
unions of k disjoint s-stars	$\sqrt{\log n}$	$\sqrt{\log n}$
$\sqrt{s} \times \sqrt{s}$ submatrices	$\sqrt{\frac{1}{\sqrt{s}}\log n}$	$\sqrt{\frac{1}{\sqrt{s}}\log n}$

$\max_{S\in\mathcal{C}} \sum_{i\in S} X_i$ does the job (in the sparse regime).

See e.g. Lugosi et. al. (2010): On combinatorial testing problems; Arias-Castro et. al. (2011): Detection of an Anomalous Cluster in a Network

Learning to learn



- How can we take advantage of feedback?
- How much can we gain?

Framework

The unknown signal and the goals are the same as before.

Measurement model:

$$Y_t = x_{A_t} + W_t, \ \ W_t \overset{iid}{\sim} N(0,1), \ \ t = 1, \dots, n \ ,$$

where $A_t \in \{1, \dots, n\}$ can depend on past observations $\{A_j, Y_j\}_{j=1}^{t-1}$.

Framework

The unknown signal and the goals are the same as before.

Measurement model:

$$Y_t = x_{A_t} + \Gamma_t^{-1/2} W_t, \;\; W_t \overset{iid}{\sim} N(0,1), \;\; t = 1, 2, \ldots \;\; ,$$

where $A_t \in \{1, \ldots, n\}$, $\Gamma_t > 0$ can depend on past observations $\{A_j, \Gamma_j, Y_j\}_{j=1}^{t-1}$, and must satisfy

$$\mathbb{E}_S \left(\sum_t \Gamma_t \right) \leq n, \; \forall S \in \mathcal{C} \; .$$

Simple procedure for recovery

Let $\mathcal{C}$ be the class of all s-sparse sets and suppose we wish to recover the support (we want $\widehat{S}$ s.t. $\max_{S\in\mathcal{C}} \mathbb{P}_S(\widehat{S} \neq S) \leq \varepsilon$).

Algorithm

- Fix $\Gamma_t = \Gamma = 1/3\ \forall t \in \mathbb{N}$
- For each entry $x_i,\ \ i = 1, \ldots, n$ do the following:
  - Measure $Y_{i,j} = x_i + \Gamma^{-1/2} W_i,\ j = 1, \ldots, \tau_i$, where $\tau_i = \min\{j : Y_{i,j} \leq 0\} \wedge \log_2(n/\varepsilon)$.
- $i \in \widehat{S} \iff Y_{i,j} > 0 \forall j = 1, \ldots, \log_2(n/\varepsilon)$.

Simple procedure for recovery - analysis

For $i \notin S$

$$\mathbb{P}(i \in \widehat{S}) = \mathbb{P}(Y_{i,j} > 0 \ \forall j) \leq (1/2)^{\log_2(n/\varepsilon)} = \varepsilon/n \ .$$

For $i \in S$

$$\mathbb{P}(i \notin \widehat{S}) \leq \mathbb{P}(\exists j : Y_{i,j} \leq 0) \leq \frac{\log_2(n/\varepsilon)}{2} e^{-\mu^2/6} \leq \varepsilon/s$$

whenever $\mu \geq \sqrt{6\left(\log \frac{s}{\varepsilon} + \log \frac{\log_2(n/\varepsilon)}{2}\right)}$.

Hence

$$\mathbb{P}_S(\widehat{S} \neq S) \leq \sum_{i \notin S} \mathbb{P}(i \in \widehat{S}) + \sum_{i \in S} \mathbb{P}(i \notin \widehat{S}) \leq \varepsilon \ .$$

Simple procedure for recovery - analysis

How much precision do we use in expectation?

$$\mathbb{E}_S\left(\sum_t \Gamma_t\right) \leq \Gamma\left(\sum_{i\notin S} 2 + \sum_{i\in S}\log_2(n/\varepsilon)\right) \leq \frac{1}{3}(2n+s\log(n/\varepsilon)) \leq n$$

if $s \ll n$.

To summarize, this simple procedure succeeds when

$$\mu \gtrsim \sqrt{\log s + \log\log n + \log\frac{1}{\varepsilon}}\ .$$

Reminder - SLRT (Wald)

We wish to test $H_0: \ Y_j \sim N(0, \Gamma^{-1})$ and $H_1: \ Y_j \sim N(\mu, \Gamma^{-1})$, $j \in \mathbb{N}$ with as few observations as possible (in expectation) with prescribed error probabilities α, β. Consider the process

$$Z_0 = 0, \ Z_t = \sum_{j=1}^{t} \log \frac{f_1(Y_i)}{f_0(Y_i)} \ , t = 1, 2, \dots$$

Let $l = \log \beta < 0 < u = \log(1/\alpha)$ and $T = \inf\{t : Z_t \notin (l, u)\}$. We then have $\mathbb{P}_0(Z_T \geq u) \leq \alpha$ and $\mathbb{P}_1(Z_T \leq l) \leq \beta$.

As $\Gamma \to 0$ we also have

- $\mathbb{P}_0(Z_T \geq u) \to \alpha$ and $\mathbb{P}_1(Z_T \leq l) \to \beta$
- $E_0(T) \approx \frac{2}{\Gamma\mu^2} \log \frac{1}{\beta}$ and $E_1(T) \approx \frac{2}{\Gamma\mu^2} \log \frac{1}{\alpha}$

Refinement

Replace the core of the previous procedure with a SLRT to test between $x_i = 0$ and $x_i = \mu$. Set Type I and II error probabilities to be $\alpha = \varepsilon/n$ and $\beta = \varepsilon/s$.

We have $\mathbb{P}_S(\widehat{S} \neq S) \leq \varepsilon$ as before.

The precision used (in expectation) is

$$\mathbb{E}_S\left(\sum_t \Gamma_t\right) \leq \frac{2}{\mu^2}\left(n \log\frac{s}{\varepsilon} + s\log\frac{n}{\varepsilon}\right) .$$

If n is large (and $s \ll n$) this is at most n whenever

$$\mu \geq \sqrt{2\log\frac{s}{\varepsilon} + o(1)} .$$

This is optimal.

Detection

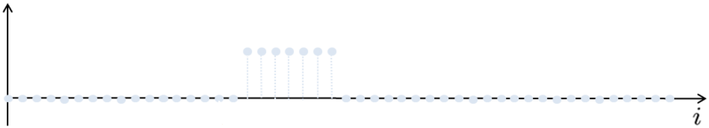
What about detection? Easy: set α as before and $\beta = \sqrt[s]{\varepsilon}$. This ensures that at least one signal component is found w.p. $1 - \varepsilon$ under the alternative.

Adaptive	Detection	Recovery
s-sets	$\sqrt{1/s}$	$\sqrt{\log s}$
unions of k disjoint s-intervals	$\sqrt{1/s}$	?
unions of k disjoint s-stars	$\sqrt{1/s}$	?
$\sqrt{s} \times \sqrt{s}$ submatrices	$\sqrt{1/s}$	?

Scaling laws do not depend on the structure anymore (as long as we have symmetry in the class)!

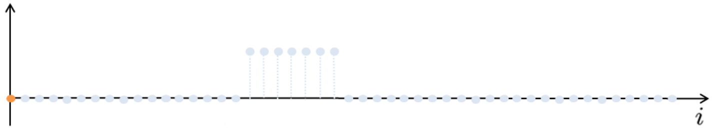
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



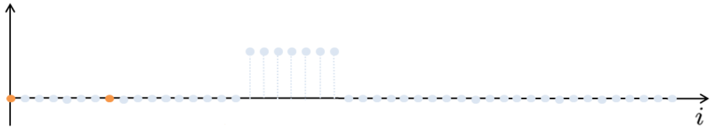
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



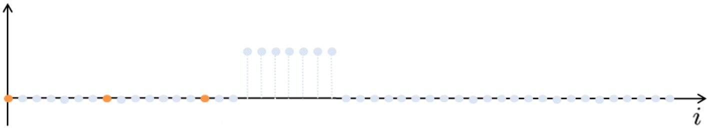
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



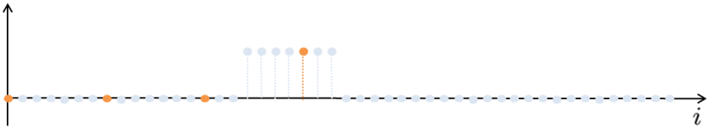
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



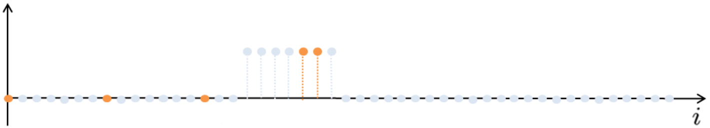
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



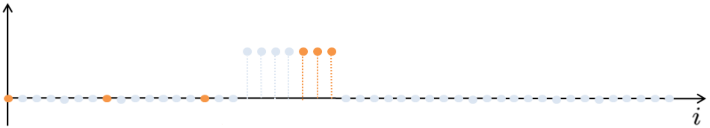
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



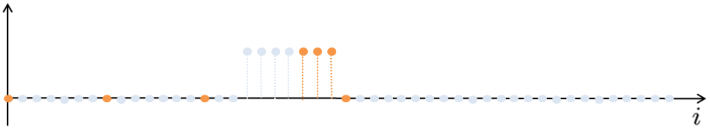
Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



Structured Recovery

For certain classes it is enough to find one component and the problem becomes "easy". For instance, if there was no noise:



Structured Recovery

Main idea: take a "noiseless case" algorithm for support recovery and "robustify" it against noise by using SLRTs.

Typically the algorithm will have two phases:

- Search: Find an active component (can also use random search)
- Refinement: Exploit structure around that component

Algorithms may alternate between the two phases (for instance in case of unions of stars).

The main difference between the two phases is that the error probabilities for the SLRTs are set differently.

Detection

Still considering probability of error we get (recall we are in the sparse regime $s \ll n$).

Adaptive	Detection	Recovery
s-sets	$\sqrt{1/s}$	$\sqrt{\log s}$
unions of k disjoint s-intervals	$\sqrt{1/s}$	$\sqrt{\frac{\log k}{s}}$
unions of k disjoint s-stars	$\sqrt{1/s}$	$\sqrt{\frac{\log k}{s}}$
$\sqrt{s} \times \sqrt{s}$ submatrices	$\sqrt{1/s}$	$\sqrt{1/s}$

Detection

For technical reasons we only managed to show lower bounds for the recovery problem considering $\max_{S \in \mathcal{C}} \mathbb{E}_S(|\widehat{S} \triangle S|) \leq \varepsilon$.

Adaptive	Detection	Recovery
s-sets	$\sqrt{1/s}$	$\sqrt{\log s}$
unions of k disjoint s-intervals	$\sqrt{1/s}$	$\sqrt{\frac{\log(ks)}{s}}$
unions of k disjoint s-stars	$\sqrt{1/s}$	$\sqrt{\frac{\log(ks)}{s}}$
$\sqrt{s} \times \sqrt{s}$ submatrices	$\sqrt{1/s}$	$\sqrt{\frac{\log s}{s}}$

Adaptive algorithms can improve on non-adaptive ones by

- Better mitigating the effects of noise $\log n \rightsquigarrow \log s$
- Better capitalizing on structure (in certain cases) $\rightsquigarrow 1/s$

Adaptive compressed sensing

Framework

The unknown signal and the goals are the same as before.

Different measurement model:

$$Y_t = < x, A^{(t)} > + W_t, \;\; W_t \overset{iid}{\sim} N(0,1), \;\; t = 1, 2, \ldots \;\; ,$$

where $A^{(t)} \in \mathbb{R}^n$ can depend on past observations $\{A^{(j)}, Y_j\}_{j=1}^{t-1}$, and must satisfy

$$\mathbb{E}_S \left(\sum_t \|A^{(t)}\|_F^2 \right) \leq n, \;\; \forall S \in \mathcal{C} \; .$$

Adaptive compressed sensing

Detection

Consider the energy test $Y_1 =< x, \mathbf{1} > + W_1$, where $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones and $\Psi = \mathbf{1}\{Y_1 > s\mu/2\}$.

We have

$$\max_{i=0,1} \mathbb{P}_i(\Psi \neq i) \leq \varepsilon ,$$

whenever $\mu \geq \sqrt{\frac{8}{s^2} \log \frac{1}{2\varepsilon}}$. This is optimal among all tests (adaptive or non-adaptive).

Structure and adaptivity do not play a role.

Arias-Castro (2012): Detecting a vector based on linear measurements

Adaptive compressed sensing

Simple procedure for recovery

Consider the 1-sparse case, and a binary search algorithm.

Let $A^{(1)} \in \mathbb{R}^n$ s.t. $A_i^{(1)} = \mathbf{1}\{i \leq n/2\}$ and $Y_1 =< x, A^{(1)} > +W_1$.
If $Y_1 > \mu/2$ "go left" otherwise "go right", and iterate.

This simple procedure has $\max_{S \in \mathcal{C}} \mathbb{P}_S(\widehat{S} \neq S) \leq \varepsilon$ whenever

$$\mu \geq \sqrt{8 \left(\log \frac{\log_2 n}{2} + \log \frac{1}{\varepsilon} \right)} \; .$$

(also $\sum \|A^{(t)}\|_F^2 \leq n$)

Similarly as before, replacing the observations by SLRTs (multiple measurements with small sensing energy) we can get rid of the $\log \log$ term.

Adaptive compressed sensing

Recovery

One can use the insights gained above for structured recovery. For s-sparse sets do s binary searches in parallel.

For structured sets do two phases as before. In the search phase

- **Intervals:** Search for a block of activation.
- **Stars:** Search for the center of the active star.
- **Submatrices:** Search for rows that contain activation.

The refinement phases are "easy" when $s \ll n$ (compared to the search phases).

Malloy, Nowak (2013): Near-Optimal adaptive compressed sensing

Adaptive compressed sensing

Recovery

Recovery	Non-adaptive	Adaptive
s-sets	$\sqrt{\log n}$	$\sqrt{\log s}$
unions of k disjoint s-intervals	$\sqrt{\frac{\log n}{s^2}}$	$\sqrt{\frac{\log(ks)}{s^2}}$
unions of k disjoint s-stars	$\sqrt{\log n}$	$\sqrt{\frac{\log(ks)}{s^2}}$
$\sqrt{s} \times \sqrt{s}$ submatrices	$\sqrt{\frac{\log n}{\sqrt{s}}}$	$\sqrt{\frac{\log(ks)}{s}}$

Non-adaptive rates are necessary, adaptive ones are sufficient and except for submatrices also necessary.

Similar behavior as before.

Remark - number of measurements

Appeal of compressive sensing: few measurements ($\approx s \log n$). We lose this in the algorithms above.

Note that in binary search $\|A^{(t)}\|_0 = 2^{-t}$. This allows us to choose $\|A^{(t)}\|_F^2 \sim t2^{-t}$ and still satisfies $\sum_t \|A^{(t)}\|_F^2 \leq n$. This way we get rid of the $\log \log$ term (at the price of an increase in the constant).

Same can be done to all other algorithms $\rightsquigarrow$ same performance, small number of measurements.

In the non-adaptive case $s \log n$ measurements are optimal. In the non-adaptive case we don't know (yet).

Final remark

The crux of all adaptive sensing algorithms is the sampling strategy.

We aim to collect the most "informative" samples based on what we already learned.

Would a sampling strategy that at time $t = 1, 2, \ldots$ decides what to do based on the posterior of $S|Y_1, \ldots, Y_{t-1}$ make sense?