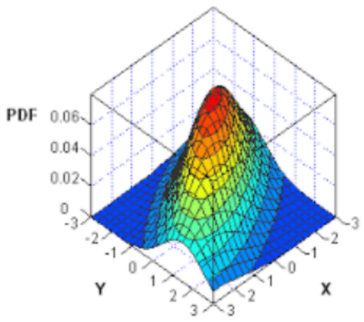
A Greedy Anytime Algorithm for Sparse PCA

Appeared in COLT 2020

Guy Holtzman, Adam Soffer, Dan Vilenchik

School of Electrical and Computer Engineering

Ben-Gurion University



Θ

Consistency

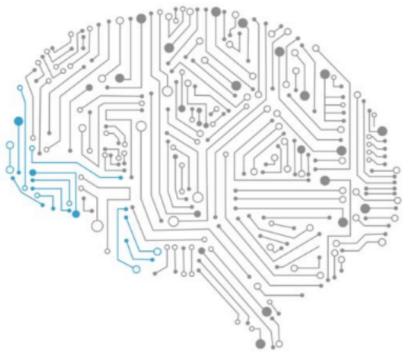
$\widehat{\Theta} \rightarrow \Theta$

$\widehat{\Theta}$

n samples from a p-dimensional distribution

Efficiency

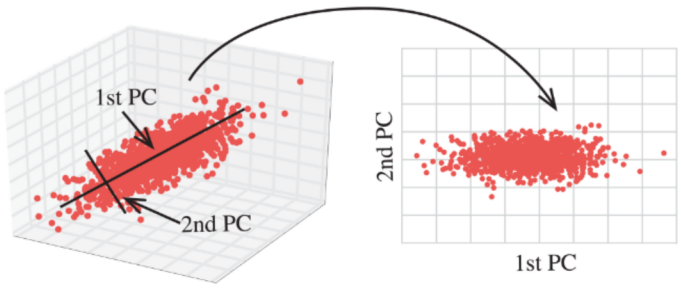
Obs	HospitalID	Sex	Cholesterol	BP_Status
1	2	Male	194	Normal
2	3	Female	200	High
3	0	Male	233	High
4	1	Female	192	Optimal
5	2	Female	209	Normal
6	3	Female	200	High
7	0	Female	184	Normal
8	1	Female	228	High
9	2	Female	150	Normal
10	3	Male	221	Normal



Estimation/Learning

Principal Component Analysis

Pearson, 1901

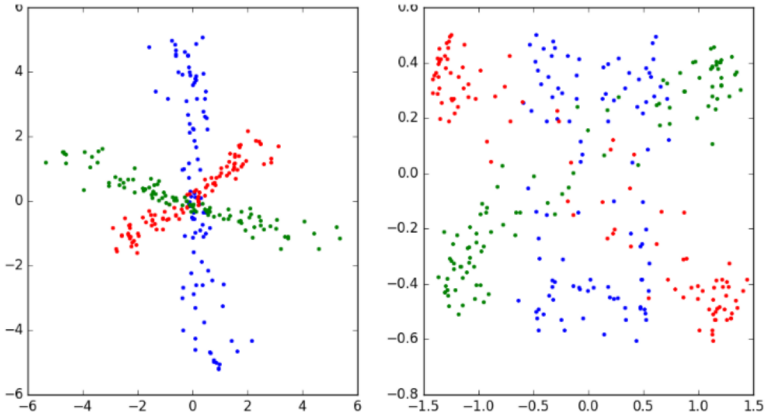


- Task: Find the direction which maximizes variance
- Reduce dimensionality
  - Computational aspect, noise reduction, visualization
- Solving PCA - eigen-decomposition of (sample) covariance matrix

$$\Sigma = \lambda_1 v_1 {v_1}^T + \lambda_2 v_2 {v_2}^T + \ldots \lambda_t v_t {v_t}^T + \lambda_{t+1} v_{t+1} {v_{t+1}}^T + \cdots + \lambda_p v_p {v_p}^T$$

PCA: The curse of dimensionality

Consistency: sample PCs $(\hat{\Sigma})$ $\overset{??}{\rightarrow}$ population PCs (Σ)?



Projection on population PCs Projection on sample PCs

- True in the classical setting $p/n \rightarrow 0$, but may not be true in high-dimensional setting (Johnstone, Paul, Bickel-Levina, Nadler)

$\hat{\Sigma} = \frac{1}{n} X^T X$, where X is the $n \times p$ design matrix

Regularization: Sparse PCA

- Assumption: leading population PC is sparse (few non-zero entries)

$$\widehat{PC}_1 = \underset{\substack{\|v\|_0 = k \\ \|v\|_2 = 1}}{\operatorname{argmax}} \mathrm{v}^T \hat{\Sigma}\, \mathrm{v}$$

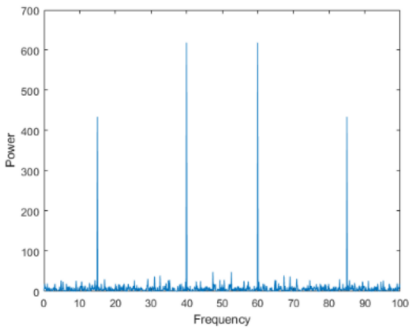
- Problem may become consistent again, $\widehat{PC}_1 \to PC_1$
- But, non-convex optimization (NP-hardness)

How to solve Sparse PCA?

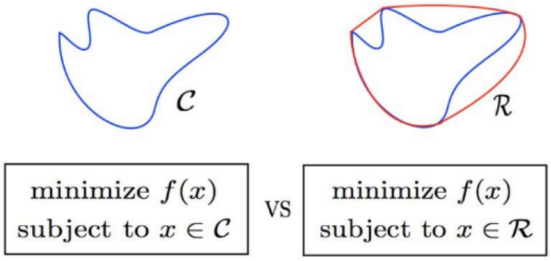
Exhaustively search all $\binom{p}{k}$ possible solutions

– Good only for small problem sizes

Polynomial time heuristics

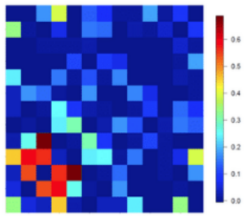


Diagonal Thresholding: Take features with largest variance (energy/amplitude) [JL09]

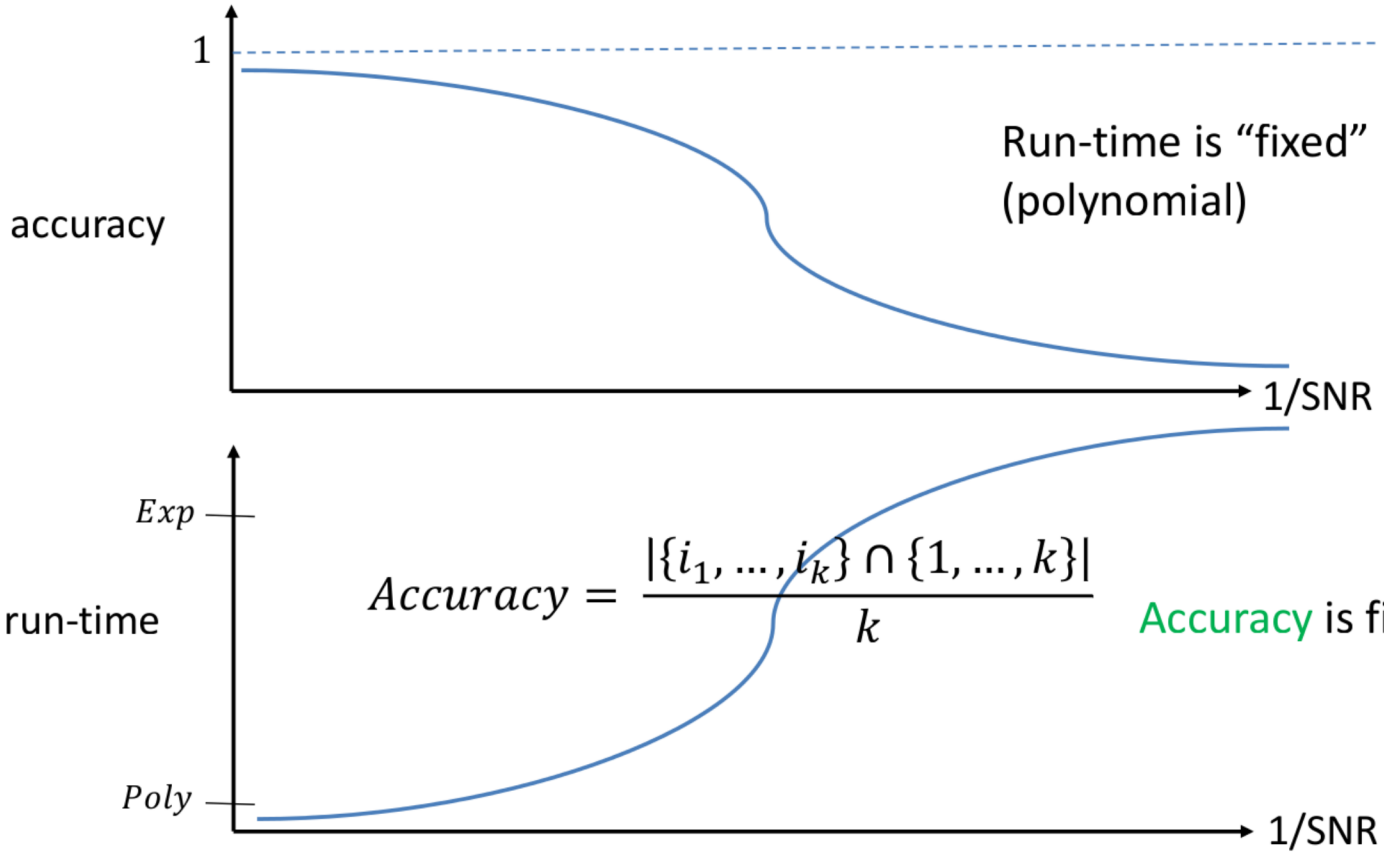


SDP Relaxation
[d`Aspremont et al. 2004]

Covariance Thresholding
[BL08,DM16]

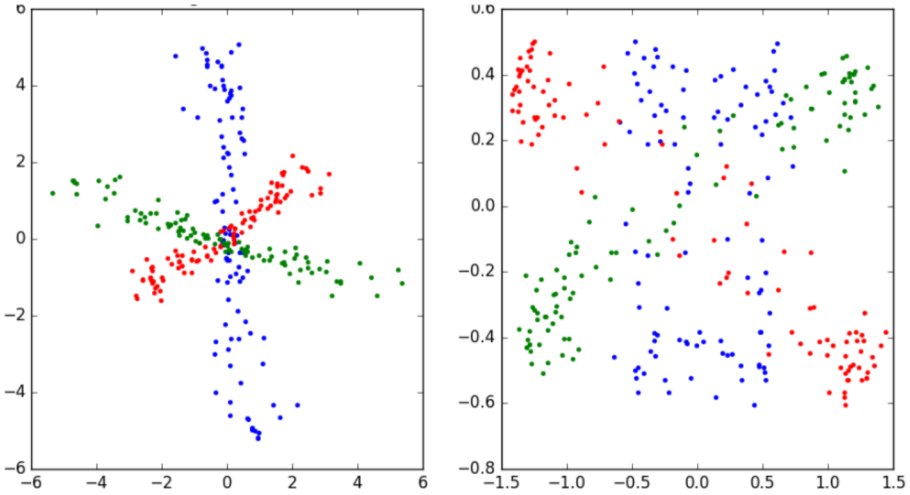


Polytime vs. Anytime



A new design principle

- Calibration of run-time to the problem's difficulty (SNR), computational resources (cluster/cloud), available time
- Here is one naïve fix: run exhaustive search with preemption
  - maximizer
- E



Seed Sparse PCA (SSPCA)

$GreedySPCA(S, k)$:

- Find $k - k^*$ variables x that maximizes $\lambda_1(\hat{\Sigma}_{S \cup \{x\}})$
- Set $S \leftarrow S \cup \{x_{i_1}, \ldots, x_{i_{k-k^*}}\}$

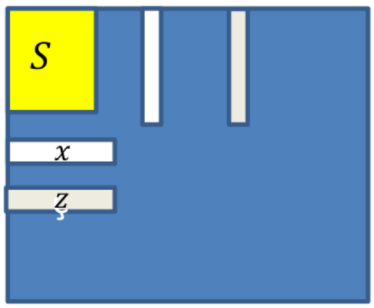
Return S

$SSPCA(k, k^*)$:

Go over all seeds S^* of size k^*

- Compute $S \leftarrow GreedySPCA(S^*, k)$

Return $\underset{S^*}{\operatorname{argmax}}\ \lambda_1(\hat{\Sigma}_S)$

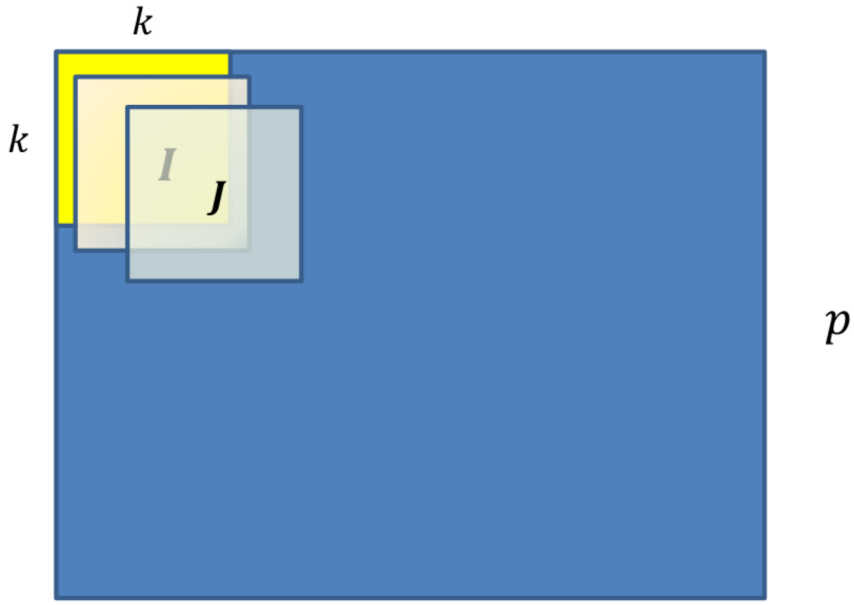


Code also available from Github https://github.com/sdannyvi/AnytimePCA/

Analysis

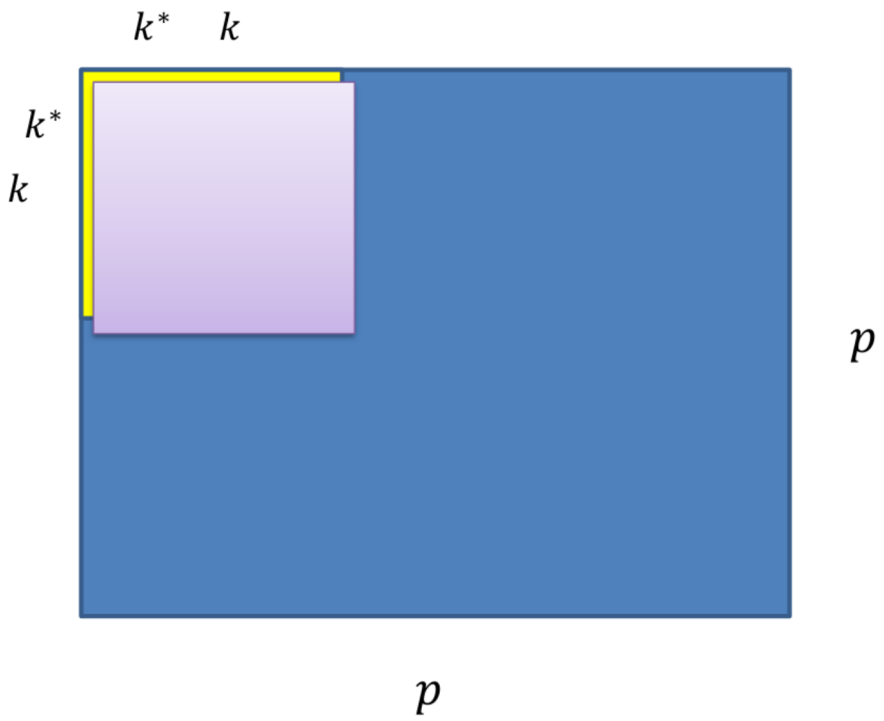
- $SSPCA$ interpolates between DT ($k^* = 0$) to Exhaustive Search ($k^* = k$)
- Running time $\binom{p}{k^*} p$ which is polynomial for $k^* = O(1)$
- Easily run in a parallel and distributed manner
- Consistency? Rigorous proof in the spiked covariance model (Johnstone 2001)
  - Spike $v^* = (\frac{1}{\sqrt{k}}, \dots, \frac{1}{\sqrt{k}}, 0,0,0.., 0)$
  - Population Covariance Matrix $\Sigma = \beta (v^*)^T v^* + I$
  - $X \sim N(0, \Sigma)$
- Similar algorithmic ideas were obtained in parallel by Ding, Kunisky, Wein, Bandeira

COND I: Spectral Separation



$\lambda_1(\Sigma_I) > \lambda_1(\Sigma_J)$ when I is "closer" to the signal than J

COND II: Golden Seeds



$\exists S$ of size k^* from which the signal is recovered

Golden Seeds

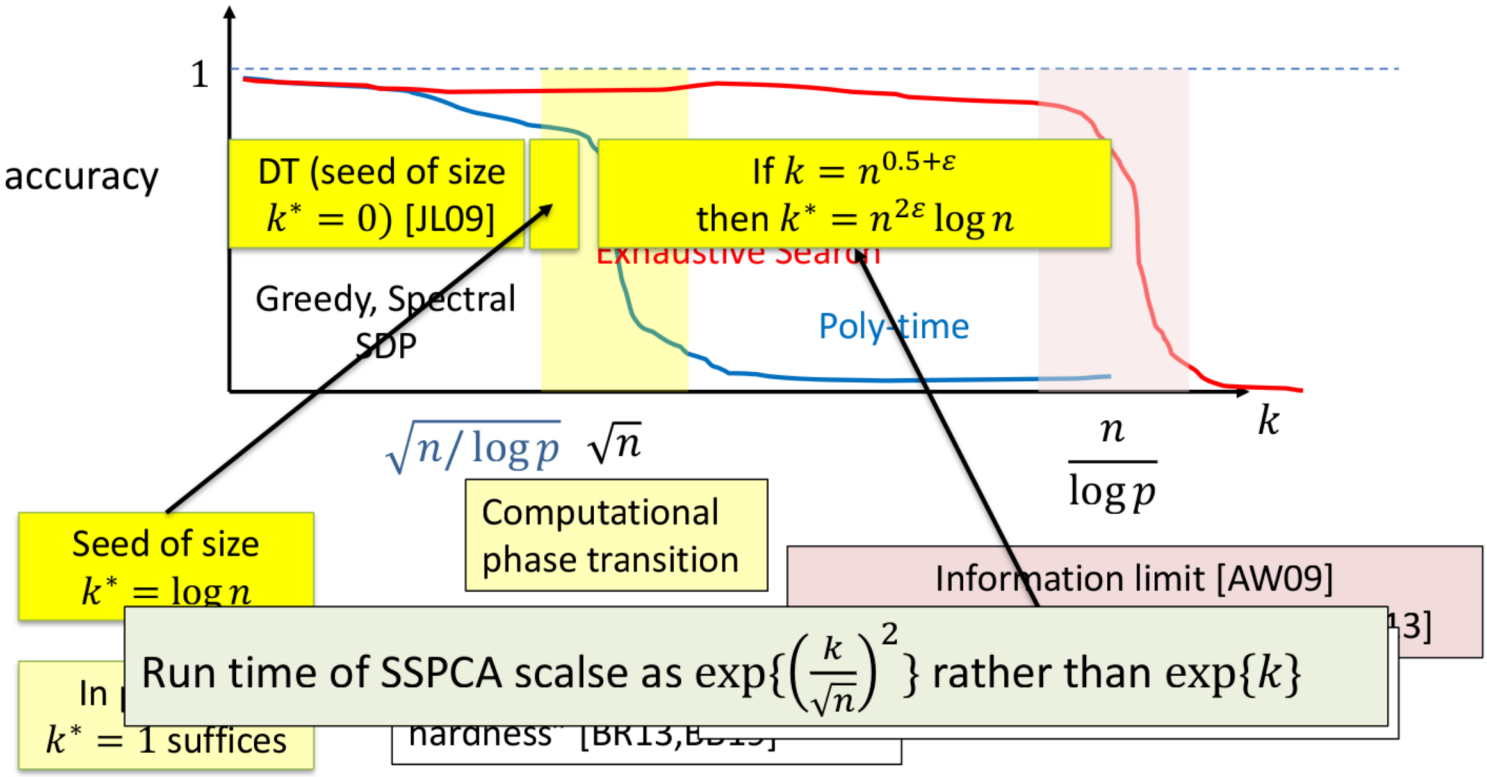
$\exists S$ of size k^* from which the signal is recovered

- Fix an initial seed S_0 to be a subset of the signal entries
- The probability of an erroneous step decreases with $k^* = |S_0|$
- Running time increase with k^*
- Find smallest k^* that allows full recovery of spike

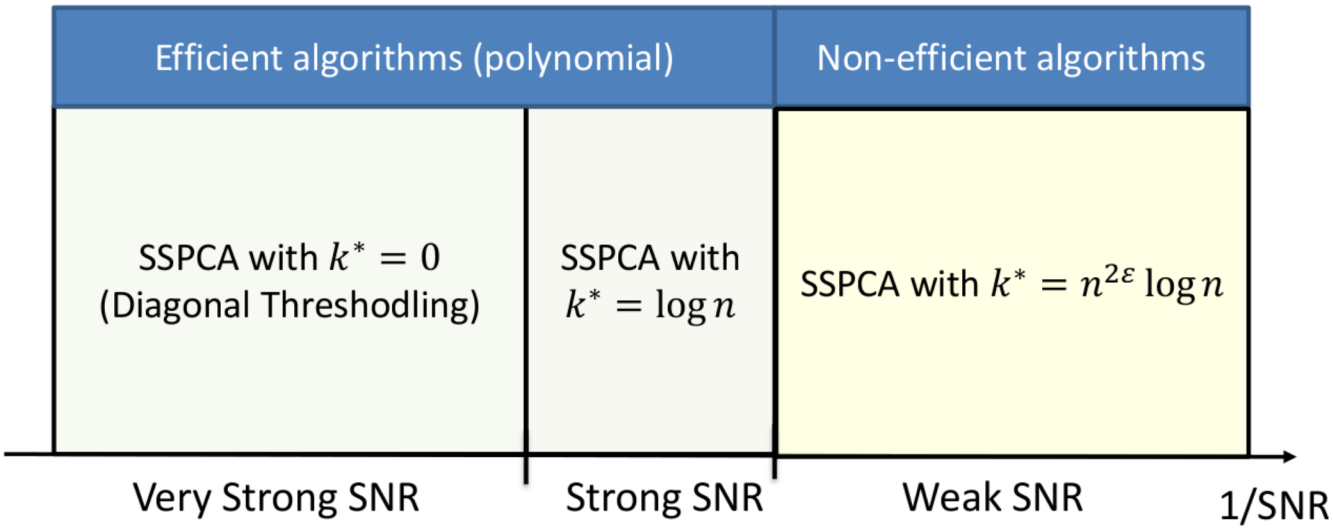
$$k^* \geq \left\lfloor \frac{Ck^2 \log n}{\beta^2 n} \right\rfloor$$

The complexity map

- k governs the SNR, and assume $n = p$



The Complexity Map Contd.



In the hard regime: run time of SSPCA scalse as $\exp\{\left(\frac{k}{\sqrt{n}}\right)^2\}$ rather than $\exp\{k\}$

Spectral Separation

For every set I of k features, If $\delta = |I \cap I^*|$, then

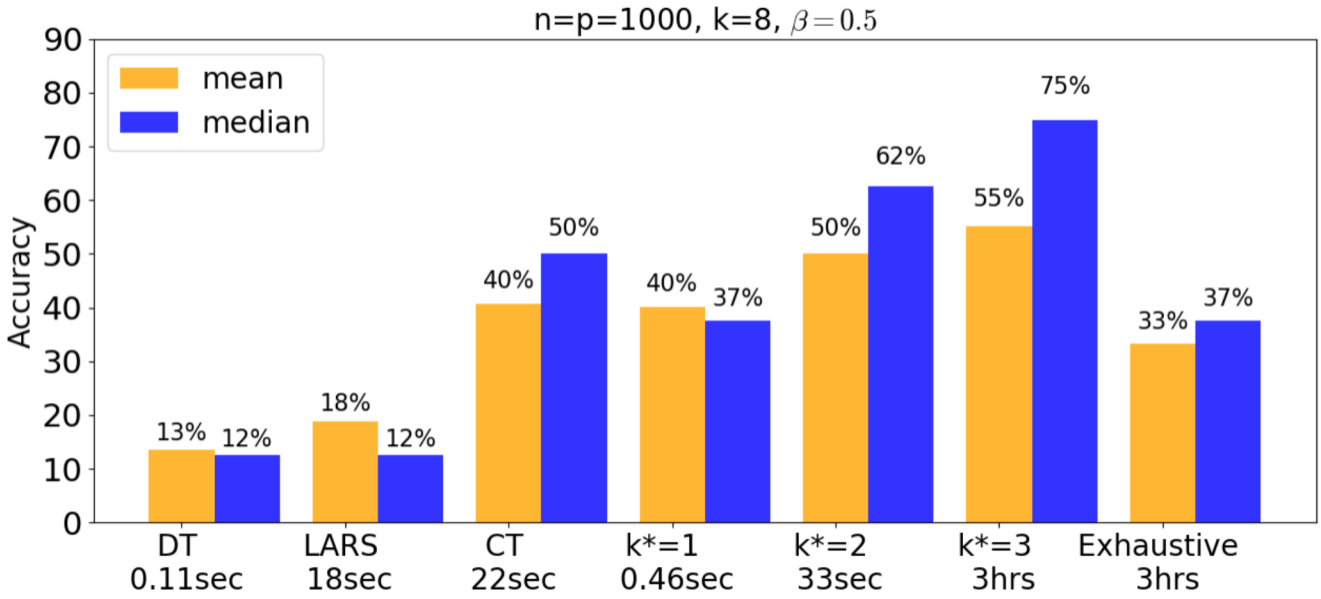
$$\lambda_1(\hat{\Sigma}_{\mathcal{I}}) \in \left[1 + \delta\beta - \Gamma, 1 + \delta\beta + \Gamma + \frac{\beta}{k}\right]$$

$$\Gamma = C\left(\frac{(1+\beta)k\log n}{n}\right)^{0.5}$$

Think of $\delta = 0.7, \beta = 0.1, k \leq n^{0.6}$

- This means that one can detect also partial solutions
- Similar results (mini-max) e.g. [Vu-Lei 2012], for δ =1

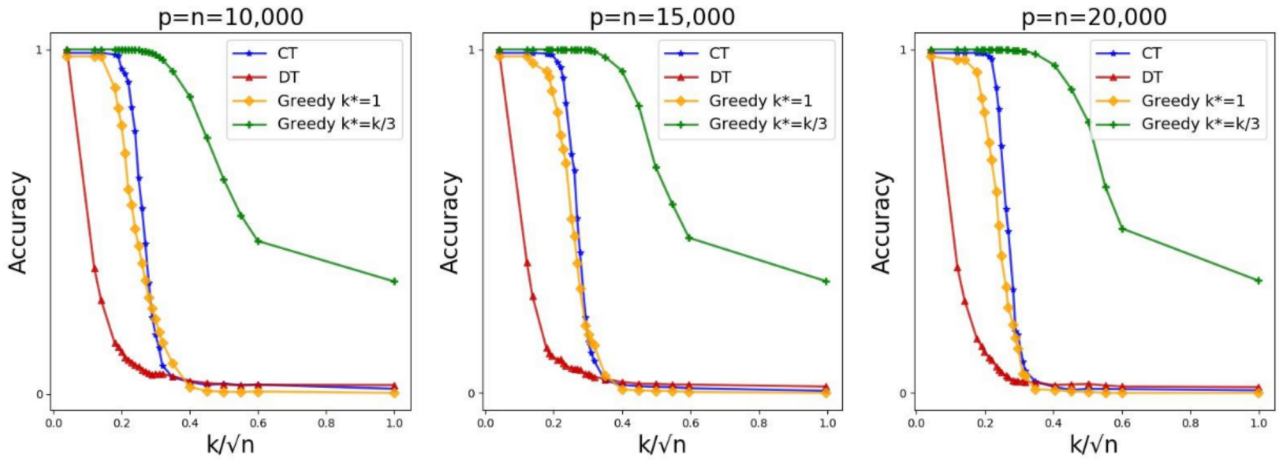
Simulation



Mean/Med over 50 executiosns using a cluster of 90 Intel Xeon Processor E7-4850 v4 (40M Cache, 2.10 GHz) cores

Warm thank-you to Johnathan Rosenblatt

Golden Seeds in the Hard Regime



Sub-modular Optimization

- Define $f: 2^{\{1,\dots,p\}} \to \mathbb{R}$

$$f(S) = \lambda_1(\Sigma_S)$$

- $f(S)$ is sub-modular if for $S' \subseteq S$ then for every $x \notin S$,

$$f(S \cup \{x\}) - f(S) \leq f(S' \cup \{x\}) - f(S')$$

- $f(S)$ is monotone if for $S' \subseteq S$,$f(S') \leq f(S)$

$$OPT = \underset{|S|=k}{\operatorname{argmax}} f(S)$$

1. $S \leftarrow \emptyset$
2. for $i = 1$ *to* k:
3. $\quad i^* \leftarrow \operatorname{argmax} f(S \cup \{x_i\})$
4. $\quad S \leftarrow S \cup \{x_{i^*}\}$

1. $S \leftarrow$ *set of size* k^*
2. for $i = 1$ *to* $k - k^*$:

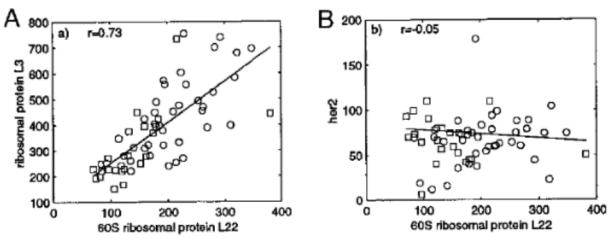
Sub-modular Optimization

- If f is sub-modular and monotone, then greedy gives a $(1-e^{-1})$-approx of OPT (Fisher, Nemhauser and Wolsey, 1978)
- Alas, this guarantee may be useless
  - Wishart distribution (white Gaussian noise), OPT$\approx$1
  - When adding a spike with energy β, OPT increases to $1+\beta$
  - $(1-e^{-1})\cdot(1+\sigma^2)<1$
- We added a seed, and an exhaustive search wrap
- Open problem: prove that the non-bulk version of SSPCA works
  - Simulation shows that both have the same performance
  - Main challenge – there are many ways of adding the correct variables, and a simple union bound kills the proof

Take Home Message/Questions

- The Anytime paradigm may be more suitable nowadays
  - Design algorithms that can take advantage of the huge amount of cheap compute resources
  - Advantageous even in “easy” regimes with small seed
- Why don’t practitioners use sparse PCA?

Dataset [Source]	No. of samples	No. of features	No. Class 0	No. of Class 1
Alon [25]	62	2,000	40	22
Tian [26]	173	12,625	36	137
Gordon [22]	181	12,533	94	87
Golub [21]	72	7,129	47	25
Burczynski [27]	127	22,283	85	42
Pomeroy [28]	60	7,128	39	21
Singh [29]	102	12,600	52	50
Chowdary [30]	104	22,283	62	42



Alon et al. PNAS, 1999

Take Home Message/Questions

- Why don't practitioners use sparse PCA?
  - n and p do not tend to infinity. If an algorithms has many tunable parameters, that depend on n, p then it's less attractive
  - t-SNE – tweak it until you get the right picture
  - Possible solution: white-box "simple" algorithms if possible
    - Instead of increasing the sophistication of the algorithm (DT < LASSO < CT < SDP), use a simple greedy algorithm and increase its running time
    - SSPCA has only one easily tunable parameter, k^*