Lecture ②

Today : Algorithms for weak recovery
(i) Spectral algorithms.
(ii) Recovery using semi-definite programming.

(i) Spectral algorithms.

Heuristics: ($\mathbb{Z}_2$-synchronization) $M = \frac{\lambda}{n}\sigma\sigma^T + W$.

$$\mathbb{E}[M|\sigma] = \frac{\lambda}{n}\sigma\sigma^T.$$

Obs: Spectral decomposition of $\mathbb{E}[M|\sigma]$ recovers σ!

$$M = \underbrace{\mathbb{E}[M|\sigma]}_{\text{signal}} + \underbrace{M - \mathbb{E}[M|\sigma]}_{\text{noise}}$$

If noise is sufficiently small, same algorithm should work!

Thm: (Féral & Péché '07, Benaych-Georges & Nadakuditi '11)
ⓐ If $\lambda \le 1$, $\lambda_{max}(M) \longrightarrow 2$ a.s. & if $\hat{v}_{PCA}$ is the top eigenvector (unit norm) then $\frac{1}{n}\langle \sigma, \hat{v}_{PCA}\rangle^2 \xrightarrow{a.s.} 0$.

ⓑ If $\lambda > 1$, $\lambda_{max}(M) \longrightarrow \lambda + \frac{1}{\lambda}$, $\frac{1}{n}\langle \sigma, \hat{v}_{PCA}\rangle^2 \xrightarrow{a.s.} 1 - \frac{1}{\lambda^2}$.

NB: a) PCA achieves the ~~PCA~~ IT threshold for $\mathbb{Z}_2$-synchronization.
b) Alg Takeaway – Just do PCA on this problem! spectral algorithms often achieve the sharp threshold.

- Spectral algorithms are often good to go. Do they always work? — No!

Community detection for sparse SBM.

$\sigma = (\sigma_i)$ $\quad \sigma_i \overset{iid}{\sim} \begin{cases} +1 & \text{wp } 1/2 \\ -1 \end{cases}$, $\quad \mathbb{P}[\{i,j\} \in E | \sigma] = \begin{cases} a/n & \text{if } \sigma_i = \sigma_j \\ b/n & \text{o.w.} \end{cases}$

Heuristics. $G \sim G(n, a/n, b/n)$ A - adjacency matrix.

$$d = \frac{a+b}{2}. \qquad \mathbb{E}[A|\sigma] = \frac{1}{n}\left[\begin{array}{c|c} a & b \\ b & a \end{array}\right] \Rightarrow \mathbb{E}[A|\sigma] - \frac{d}{n}11^T$$

$$= \frac{a-b}{2n}\sigma\sigma^T.$$

$$A - \frac{d}{n}11^T = \underbrace{\frac{a-b}{2n}\sigma\sigma^T}_{\text{signal}} + \underbrace{A - \mathbb{E}[A|\sigma]}_{\text{noise}}$$

PCA yields σ!

Guess 0: If noise is small, we will recover σ!

In fact, this naive spectral algorithm fails spectacularly!

Q. Why does the algorithm fail?

Some insights from sparse Erdos-Renyi random graphs.

Thm: (Krivelevich & Sudakov '01)

Let $G \sim G(n,p)$. A - adjacency matrix. $\frac{\lambda_{max}(A)}{\max\{\sqrt{\Delta}, np\}} \xrightarrow{P} 1.$

Δ - max degree.

NB: (i) If $p = c/n$, $np = O(1)$, $\Delta = O\left(\frac{\log n}{\log\log n}\right)$.

$$\therefore \Rightarrow \lambda_{max}(A) = O\left(\frac{\log n}{\log \log n}\right).$$

(ii) Large eigenvalues are created by vertices with large degrees. Corresponding eigenvalues are quite localized on these high-degree vertices.

(iii) Similar phenomena for SBM. Consequently, naive PCA does not work in this regime.

(iv) PCA works for denser graphs - many interesting results

Q. How to redeem spectral algorithms at this point?

→ Remove high-degree vertices

→ Use alternative spectral methods. (eg., non-backtracking matrix)

→ Use alternative algorithms, less sensitive to large degree vertices. (eg., semidefinite programming)

• How do these strategies work?

Assume $d>1$ ($\exists$ a giant comp)

Thm: $G \sim G(n, a/n, b/n)$. $\lambda = \frac{a-b}{\sqrt{2(a+b)}}$. If $\lambda \leq 1$, weak recovery is impossible. If $\lambda > 1$, weak recovery is possible.

~~Det~~

• Remove high-degrees vertices (weak recovery for $\lambda > C$ sufficiently large)

→ Unclear if this achieves $\lambda = 1$.

→ Threshold is also subtle.

Spectral algorithm based on non-backtracking matrix.

Defn: $G = (V, E)$ $\quad B \in \mathbb{R}^{2|E| \times 2|E|}$ $\quad$ If $e\{i,j\} \in E$, add oriented edges $e = (i,j)$ & $e^{-1} = (j,i)$

$$B_{ef} = \mathbb{1}(e_2 = f_1)\,\mathbb{1}(e_1 \neq f_2)$$

(this is a non-symmetric matrix)

Thm: (Bordenave, Lelarge, Massoulié '15) (informal)

The top eigenvector of B allows weak recovery.

The modified spectral algorithm works in this context.

NB: A modified spectral algorithm works in this context.

(Discuss intuition)

Convex relaxation based approaches (based on Semi-definite programming (SDP)).

Approach ① Convex relaxation of MLE.

Ex ($\mathbb{Z}_2$-synchronization) $M = \frac{\lambda}{n}\sigma\sigma^T + W$.

$$\hat{v}_{MLE} = \underset{v \in \{\pm 1\}^n}{\operatorname{argmax}}\, v^T M v.$$

$\left[\langle A,B\rangle = \mathrm{Tr}(AB^T)\right]$

$\hat{v}_{MLE}$ solves $\max v^T M v = \max \langle M, vv^T\rangle$

$$= \max \langle M, X\rangle \text{ subject to } X \succcurlyeq 0, \; X_{ii} = 1 \;\forall i \in [n], \; \mathrm{rank}(X) = 1.$$

Idea (i) Solve $\max \langle M, X\rangle$ subject to $X \succcurlyeq 0$, $X_{ii} = 1 \;\forall i \in [n]$.

(ii) Round the solution $\hat{X}$ to get an estimator $\hat{\sigma}_{SDP}$.

Theorem (informal)

For $\lambda > 1$, $\hat{\sigma}_{SDP}$ achieves weak recovery.

For community detection $\hat{\sigma}_{SDP}$ works for any $\lambda > 1$, provided the average degree (d) is large enough.

[SDP for block model: $\max \left\langle A - \frac{d}{n}11^T, X\right\rangle$ s.t. $X \succcurlyeq 0$, $X_{ii} = 1 \;\forall i \in [n]$.]

NB: This SDP cannot work as soon as $d > 1$.

Heuristic: $\lambda = 0$. $\frac{1}{n} \max_{\substack{X \succeq 0 \\ X_{ii} = 1}} \langle W, X \rangle \le \frac{1}{n} \max_{\substack{X \succeq 0 \\ Tr(X) = n}} \langle W, X \rangle \le \lambda_{max}(W) = 2.$

In fact, $\frac{1}{n} \max_{\substack{X \succeq 0 \\ X_{ii} = 1}} \langle W, X \rangle \xrightarrow{P} 2.$

Approach ② The meta approach of Hopkins-Steurer '17.

Meta-algorithm

a) Construct a low degree polynomial in the data entries to estimate $\sigma\sigma^T$.

b) Apply a robust eigenvector/SDP based algorithm to construct $\hat{\sigma}$ from $\widehat{\sigma\sigma^T}$.

Example (Community detection for SBM) $\sigma \in \{\pm 1\}^n$

$$\mathbb{P}[\{i,j\} \in E \mid \sigma] = \begin{cases} a/n & \text{if } \sigma_i = \sigma_j \\ b/n & \text{if } \sigma_i \neq \sigma_j \end{cases}$$

Define $d = \frac{a+b}{2}$

$\varepsilon = \frac{a-b}{a+b}$.

$\varepsilon d = \frac{a-b}{2}$.

Step ① Construct a low degree polynomial estimator for $\sigma\sigma^T$.

$\cancel{\mathbb{E}[p_{ij}(A) \mid \sigma] = \frac{a}{\varepsilon d}}$

$$p_{ij} = \frac{n}{\varepsilon d}\left(A_{ij} - \frac{d}{n}\right)$$

$$\mathbb{E}\left[A_{ij} - \frac{d}{n} \,\middle|\, \sigma\right] = \cancel{\frac{a}{n}\left(1 - \frac{d}{n}\right) + \frac{b}{n}\left(-d/n\right)} \begin{cases} \frac{a}{n}\left(1-\frac{d}{n}\right) + \left(-\frac{d}{n}\right)\left(1 - \frac{a}{n}\right) \\ = \frac{a-d}{n} & \text{if } \sigma_i\sigma_j = 1 \\ = \frac{b-d}{n} & \text{if } \sigma_i\sigma_j = -1 \end{cases}$$

$$= \left\{ \frac{a-b}{2n} \sigma_i \sigma_j \right.$$

$\therefore \frac{2n}{a-b}\left(A_{ij} - \frac{d}{n}\right)$ is unbiased for $\sigma_i \sigma_j$.

$$\mathrm{Var}\left(\frac{2n}{a-b}\left(A_{ij}-\frac{d}{n}\right)\Big|\,\sigma\right) = \begin{cases} \frac{4n^2}{(a-b)^2}\cdot\frac{a}{n}\left(1-\frac{a}{n}\right) & \text{if } \sigma_i = \sigma_j \\ \frac{4n^2}{(a-b)^2}\cdot\frac{b}{n}\left(1-\frac{b}{n}\right) & \text{if } \sigma_i \neq \sigma_j. \end{cases}$$

$$\mathbb{E}\,\mathrm{Var}\left(\frac{2n}{a-b}\left(A_{ij}-\frac{d}{n}\right)\Big|\,\sigma\right) = O(n) \text{ (very high!)}$$
$$= O\left(\frac{n}{\varepsilon^2 d}\right)$$

Idea:

Let α be a path (on the complete graph) between i & j

Consider the estimator $p_\alpha(x) = \prod_{ab \in \alpha} p_{ab}(x)$.

$$\mathbb{E}\left[p_\alpha(x)\,|\,\sigma\right] = \mathbb{E}\left[\prod_{ab\in\alpha} \sigma_a \sigma_b \,\Big|\, \sigma\right] = \sigma_i \sigma_j.$$

$$\mathbb{E}\left[\mathrm{Var}\left(\mathbb{E}\left[p_\alpha(x)\,|\,\sigma\right]\right)\right] \approx \left(\frac{n}{\varepsilon^2 d}\right)^{l}. \text{ (still large!)}$$

Idea: Avg. over these paths. (approximately n^{l-1} many)

If walks roughly ind, the avg. estimator has variance $\frac{n}{(\varepsilon^2 d)^l}$. If $\varepsilon^2 d \geq 1+\delta$, can choose $l = c\log n$

to get consistency.

Fact: Need self-avoiding walks to get approximate independence.